FLEXIBASE/IMAGE

Technical Reference Manual

26th Edition
December 1998

The FLEXIBASE/IMAGE Software

The **FLEXIBASE** software is distributed by the Open Seas SA.

Copyright Notice

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

SECTION 1 Ȼ OVERVIEW

INTRODUCTION

FLEXIBASE provides a comprehensive set of facilities for the management and analysis of TurboIMAGE databases on HP3000 computers. It provides systems development staff with database restructuring and test database creation functions. It provides operations staff with quick capacity changes, automatic capacity management and database repair functions. It provides system managers with database performance analysis and optimization tools. It provides computer auditors with integrity checking and database transaction analysis facilities. The functions listed below are selectable from a menu.

DIAGNOSE Function

This facility provides statistics on the contents of a database and other information that is useful for analysing database performance. Because it contains a rule-based expert system, it will automatically recommend improvements that can be made to a database such as reblocking, detail set repacking, or capacity changes where appropriate.

The DIAGNOSE function only performs a serial read of the sets selected for analysis and it therefore runs quickly. At the same time as collecting the necessary statistics, it also performs some checks on the format and integrity of the database including a check on the correctness of detail and master set chains.

The DIAGNOSE function is best used to regularly scan databases to identify faults or to report on improvements that can be made in performance. Because it can scan many databases and analyze the statistics for you, even database experts can save a lot of time and effort with this facility.

CHECK Function

The CHECK function enables you to check that a database adheres to the rules required by the application design. It can check for logical data relationships which are not enforced by the way the application design is mapped onto the database design (in addition to the master-detail referential integrity checks imposed by IMAGE).

For example: it can check that there are matching records on two datasets for a given item value, when those two records are not linked by the normal master-detail relationship. It can also check many other useful parameters, such as lengths of chains, item values totalling correctly and that the data item values in a set are unique (for example, that Invoice Numbers, Product Codes or Order Numbers are not duplicated).

By using this facility on a regular basis, it is easy to pick up logical integrity errors in the database caused by program errors, design defects, the misuse of QUERY, malicious corruptions, software aborts and many other sources of data error.

ANALYZE Function

The ANALYZE function reads and analyzes IMAGE transaction log files. It can be used to provide answers to "who did what" questions and as a general transaction audit facility. It also provides statistics that are useful for database performance analysis.

Extensive enquiry and reporting options are available to the user so that only the relevant log records can be selected for analysis. For example selection can be on the basis of data set accessed, user program, terminal device in use, data item changed, etc.

Comprehensive statistics on the actual usage of a database are provided which are stored and accumulated in a history file. Those datasets that receive the most write access can be identified as well as the active programs or users. Information collected by the ANALYZE function is also used in the DIAGNOSE module.

CAPACITY Function

This module provides quick capacity changes on master or detail data sets. It does in seconds or minutes what would otherwise take hours or days with a full DBUNLOAD/DBLOAD. If there is not sufficient work space on disc it will operate via a tape work file (unlike other products which would fail in this circumstance).

AUTOMATIC Capacity Function

This module provides for automatic management of data set capacities. You can create a job that scans a range of databases (or data sets) and compares them against user defined parameters. These parameters are: the maximum percentage full to be permitted, the target percentage full and the minimum percentage full. If the actual number of entries exceed the maximum, a capacity change will be invoked by the job to reduce the percentage full to the target. Likewise if the actual number of entries is less than the minimum, a capacity change will be invoked to increase it to the target. If such a job is run regularly then the risk of exceeding data set capacities can be avoided.

A formula for the target capacity can also be specified; for example based on the entries in other sets or the historic maximum number of records (the "High Water Mark").

REPACK Function

This feature enables a detail data set to be reloaded in primary path sequence so that speed of access to the data set is optimized - this can improve application response times and throughput.

REPAIR Function

The purpose of this module is to correct any broken chains that have arisen in access paths to detail data sets. Broken chains can arise occasionally due to a variety of hardware and software problems.

TRANSPOSE Function

This function enables the value of a data item to be globally changed throughout an existing IMAGE database. The change will be made irrespective of whether the item is an ordinary item, a sort item or a search item. When an item is a search item it is impossible to make such a change in QUERY and it is tedious to have to write a program to make such changes.

CREATE Function

The CREATE function can be used to create a new, cut-down version of a database. It copies a database but selects a limited number of records. Records can be selected on a random basis (e.g. 20% of all records) or on the basis of a specific value (or range of values).

All associated records in other data sets are copied so the new database is a logical subset of the old one. This function is particularly useful for creating small test databases but it also has other applications in the manipulation of operational databases.

UNLOAD Function

The UNLOAD function can be used to transfer the contents of a specified data set to an MPE file, i.e. the data in the data set is extracted and placed into a "flat" file. This function is useful where the data requires editing (possibly before reloading into the database).

ERASE Function

The ERASE function removes all records from a data set and leaves the data set empty. It does this in a fraction of the time that QUERY or other utility programs take to delete all records.

LOAD Function

The LOAD function can be used to load a data set from an MPE file - this file will typically have been created by an earlier use of the UNLOAD function.

RENAME Function

FLEXIBASE can rename an existing database in seconds.

COPY Function

FLEXIBASE can copy an entire database. The new database can be placed in a different group or account and can also be renamed. Possible applications are for system testing (e.g. parallel running of software) or for use as a back-up method.

DECOMPILE Function

This feature will generate a new schema file from an existing database by inspection of the database root file. It is useful where the schema has been lost.

TRANSMUTE Function

This function enables an existing database to be transformed into a new structure or format. The changes that can be made include the following:

- Changing the type or length of a data item.
- Changing the position of data items within an existing data set.
- Adding or deleting items in existing data sets.
- Changing the position of sets within a database.
- Adding or deleting data sets.
- Renaming data sets or data items.
- Adding or deleting access paths.
- Changing the capacity of a data set.
- Changing database passwords or blocking factors.

All possible changes that may be required to an IMAGE or TurboIMAGE database are incorporated. This includes changes that cannot be done with standard Hewlett Packard DBUNLOAD/DBLOAD utility programs or other database management products.

TRANSMUTE automatically identifies the need for data conversion and will retain the meanings and values of all data present in the database during the conversion. FLEXIBASE prompts the user for information such as whether to left or right justify items that have been changed in size.

TRANSMUTE performs the database operations directly on the existing database. All changes are done in one operation and only those data sets that are affected by the changes are processed - this means that changes are performed in a fraction of the time it takes using standard HP utilities or other third party products.

TRANSFER Function

This function is like the TRANSMUTE function but in this case the data is moved from one database to another at the same time as the data is changed. The changes made are determined by the differences between the originating and receiving databases. The function can be used for creating a new version of a database while retaining the original database.

ARCHIVE Function

The ARCHIVE function can be used to extract subsets of entries in a database into a smaller, archive version of the database. It selects a limited number of records from a source database, puts them into an archive database and deletes them from the original database. Records are selected on the basis of specific item values, or range of values. All associated records in other data sets will be archived so the new database will be a logical subset of the old one. This function is used to archive logical subsets of data, for example all transactions dated last year or all entries relating to a defunct company etc.

MPE Function

This function enables MMPE commands to be entered an executed (such as SHOWJOB etc.). It also enables the running of other programs such as QUERY, EDITOR or other utilities directly from the main FLEXIBASE menu.

SUPPORT FOR IMAGE/SQL

To configure an IMAGE database for SQL access, it is necessary to ATTACH the database to a DBEnvironment. Normally, it is not possible to make changes to the database structure without first DETACHing it from the DBEnvironment. The FLEXIBASE TRANSMUTE function handles the process of DETACHing and re-ATTACHing the database automatically, at the same time rebuilding all the associated SQL structures. Changes that do not affect the database structure (Capacity Change and Dataset Repacking) do not require the database to be DETACHed, and are handled transparently.

SUPPORT FOR THIRD PARTY INDEXING (TPI)

All functions within FLEXIBASE will now detect third party indexing if it is enabled, and make calls to either Omnidex or SUPERDEX to resynchronize indexes after any processing which makes this necessary. This is available for versions of Omnidex from 3.02.07, and for versions of SUPERDEX from 4.1.38. Details of how each function applies reindexing can be found in the appropriate sections.

SUPPORT FOR B-TREE INDEX

This release adds B-TREE support to all relevant Flexibase modules.

ON-LINE HELP AND SET INFORMATION

You can obtain assistance before replying to any prompt within FLEXIBASE by simply entering a question mark. You can also obtain lists of databases in the local group, lists of sets within a database, or lists of items within a set whenever required. The set display also gives a handy summary of the set capacities and percentage full and this can be obtained for any database directly from the main FLEXIBASE menu by using the SHOW function.

DATA BASE ENQUIRY/REPORTING

An optional module of FLEXIBASE is a powerful database enquiry and reporting tool named the FANTASIA REPORTER. From an easy to use menu, you can list, sort, total and otherwise report on data in an IMAGE database. This either runs interpretively, or (if you have a COBOL compiler) generates high-performance compiled programs.

SECURITY, CONTROL AND REPORTING

FLEXIBASE operates within the security constraints of IMAGE and the MPE operating system. It will only allow database access and update capability to those users with adequate authority. All functions provide a complete audit trail of all changes made to a database including an automatic print out of the activity performed (this can optionally be suppressed if not required). The progress of functions is also reported every few seconds including an indication of the percentage complete. This is done either to the users terminal or to a monitor file if running in job mode.

PERFORMANCE TRENDING

The DIAGNOSE and ANALYZE functions will load performance-related data into a central statistics database. The FLEXIBASE/IMAGE Trends module, run under MS-Windows, will allow the system manager to produce graphs showing the evolution of database performance, or to load the raw data into PC-based spreadsheets or databases.

A GENERAL SOLUTION

FLEXIBASE offers a comprehensive solution to the problems of database management. It provides facilities needed by system developers, operations managers, auditors, programmers, systems performance specialists and other staff. It can replace the ad-hoc methods or specific tools that were previously used with an integrated, comprehensive software product that is also easy to use.

FLEXIBASE requires no new command language to be learned and you can have it working within minutes with no special training requirements. Even non computer experts can use the product to optimise database performance.

SOFTWARE & HARDWARE ENVIRONMENT

FLEXIBASE is designed for use on IMAGE or TurboIMAGE data bases on the Hewlett Packard 3000 range of computers running the MPE/V or MPE/iX operating systems. On MPE/iX systems the software uses Native Mode code. Note that a complementary product named FLEXIBASE/SQL is available that provides similar facilities for ALLBASE/SQL users.

SOFTWARE INSTALLATION

Installation of the software is performed by simply restoring the software and associated files from tape and the system is then ready for immediate use. Refer to Appendix A of this manual for information on installing the software and using the demonstration version of the software. The PC software is installed using the standard MS-Windows Setup system as described in Appendix A.

DOCUMENTATION

There are two manuals provided with the FLEXIBASE/IMAGE software. These are the Technical Reference Manual (this one that you are reading) and a Guide for the New User. Refer to the latter document for an introduction to IMAGE and the use of the FLEXIBASE software. The PC software is supplied with a complete, context-sensitive help system.

SECTION 2 ₵ SHOW FUNCTION

The SHOW Function enables you to quickly and easily display a list of the sets of a database with some relevant information about each set. An example of the information displayed is as follows:

```
     Dataset Name          Type      Cap   Entries %Full

     CUSTOMER               M       20031    15008  75%
     PRODUCT                A       19035     6786  35%
     SALES                  D       35000    30076  86%
```

The column headed "Type" contains "M" for Manual Master sets, "A" for Automatic Master sets and "D" for Detail sets. The column headed "Cap" shows the current set capacity, i.e. the maximum number of entries (records) that can be put into the set. The column headed "Entries" shows the existing number of entries (records) in the set. The column headed "%Full" expresses the existing number of entries as a percentage of the capacity.

The only pre-conditions or requirements to be able to run this function are that you have read access to the database and know a database password. It will prompt you for the database name and a password (if you are the database creator but do not know a password then simply enter a password of ";"). Note that any sets that you do not have access to, with the password you supply, will not be listed.

The following is the dialog between the user and the FLEXIBASE software to run the SHOW Function:

```
    :RUN FLEXIBAS.PUB.HQ
```

Select option 1 from the menu.

```
Database name >>
```

Enter the name of the database for which the information is required.

You can also obtain a similar display of the sets within a database at any point in the FLEXIBASE software by entering a question mark in response to the prompt for a data set name (see Appendix E).

SECTION 3 ₵ DIAGNOSE FUNCTION

3.1 PURPOSE AND USER DIALOG

This module performs an examination of databases or individual datasets. It reports statistics relevant to the analysis of database throughput and user response times, and recommends changes which could improve those areas. Three forms of output are produced:

- Detailed statistics for each dataset.
- A list of recommendations for any sets needing attention.
- Comma-separated Variable files which can be downloaded into any Lotus-compatible spreadsheet (see FXUTIL documentation in section 24 for details).

Any database which is in constant use requires some form of regular monitoring. Without this datasets will fill up, performance will degrade and structural defects may remain undetected. DIAGNOSE was designed to provide users with a tool to detect those inefficiencies and defects, as well as predicting potential capacity problems. It is recommended to run DIAGNOSE on all your production databases once a week. Further information can be gathered by running the ANALYZE module at the end of each logging cycle (if logging is enabled).

Running the DIAGNOSE module regularly on your production databases will ensure close monitoring of their performance. Any recommendations for improvements can be actioned by running the TRANSMUTE function (see Section 18), CAPACITY CHANGE function (see Section 6), REPACK function (see Section 8) or the REPAIR function (see Section 9).

A utility program called FXUTIL is provided to configure the parameters and output from the DIAGNOSE and ANALYZE modules. See section 24 for details of the parameters which may be configured for the DIAGNOSE module.

```
Requirements & Pre-conditions:
•   The user requires read access to the database.
•   The user must be able to open the database in mode 5 or 6.
```

The following is the dialog between the user and the FLEXIBASE software:

```
    :RUN FLEXIBAS.PUB.HQ
```

Select option 2 from the menu.

```
    Database name >>
```

Enter the name of the database for which the statistics and diagnoses are required. The user can open any database on the system, with due regard to normal MPE security. If the user is not the creator he/she will be prompted for the database password and the program will attempt to open the database in either mode 5 or 6. Note that the DIAGNOSE function will run faster and supply more information if it has exclusive access to the database.

```
    Enter Set Name (or @ for all sets) >>
```

Enter the name of an individual set within the named database, or enter @ to diagnose ALL sets within the database in order of their appearance in the set list. Note that if you want to be more selective, you can also enter @M to diagnose all manual master sets, @A to diagnose all automatic master sets, or @D to diagnose all detail sets in the database in response to this prompt.

```
Do you want the summary or detail report on $STDLIST (S/D)? >>
```

The detail report contains all the statistics for the database, and each set, along with any recommendations. The summary report lists only the set names and any recommendations for those sets that are non-optimal. You can select which of these formats will appear on your $STDLIST device. If summary is selected then the detail report will be directed to the file FXREPORT (see below) and a copy of the summary report to file FXLIST. If detail is selected then nothing is sent to FXREPORT, but the summary report still goes to FXLIST.

```
Enter Lotus "Comma-Separated-Variable" File Name >>
```

This question will only appear if Lotus compatible output has been configured for DIAGNOSE, using FXUTIL (see section 5.8 for further details).

```
Return to continue (E exits) >>
```

If the user selects the detailed report be displayed on $STDLIST and he is running interactively, then he is prompted as to whether to continue after the information on each set is displayed. This is designed to allow the user time to examine the information before continuing. Pressing Return causes the next set to be displayed. Entering "QUIT", "EXIT", "E" or "ABORT" terminates the DIAGNOSE run.

If a summary report for all sets has been requested there follows a list of each set in order, along with an recommendations for improvement for any sets which require changes. If no significant inefficiencies have been detected then only the set names and numbers will be displayed. The following is an example of the display that is output to the STDLIST device (e.g. the users terminal):

```
-------------------------------------------------

1) Set (Automatic) CLOCK-NOS:
2) Set (Manual)    DEPARTMENTS:
   RECOMMEND ℭ Increase BLOCKING FACTOR to 11 to
   optimize buffer usage
3) Set (Detail)    EMPLOYEES:

-------------- DIAGNOSIS  COMPLETE ---------------

Further sets.....(or press return to exit)....
```

If an individual set was selected then the program prompts for further sets within the database. Press return to close the selected database.

```
Further databases.....(or press return to exit)...
```

More than one database can be examined in one run of the program. Enter the name of the next database or press return to end the program run.

3.2 PROGRESS REPORTING

In order to keep the user aware of how far through a set DIAGNOSE has progressed, it displays messages on $STDLIST (e.g. the users terminal when running in interactive mode). A number is displayed which represents the percentage complete for this dataset. These messages are displayed every 30 seconds (unless specified otherwise – see Appendix C). In addition, a file named FXPROGRn (where n is a single digit in the range 1 to 9) is created which contains the same information, along with the times at which each set was started. Normally this file will be called FXPROGR1, but if another user is already using DIAGNOSE in the same logon group then the file will be FXPROGR2 or 3 etc. If you are running DIAGNOSE in job mode you can read this file to see how far the run has progressed (e.g. by FCOPYing it to your terminal).

3.3 EXPLANATION OF STATISTICS REPORT

An example of the detail FXREPORT format is shown hereafter, followed by explanations and definitions of each statistic:

```
<><> DIAGNOSE/3000 <><>      Version D.00.01 (WED, JAN 22, 1992,  4:55 PM)

Database name >> EXBASE
Enter Set Name (or @ for all sets) >> @
Do you want the summary or detail report on the screen (S/D)? >>D
```

```
DATABASE STATISTICS AND DIAGNOSIS FOR EXBASE.DAVE.PROGDEV     Page:1

Global Statistics:
IMAGE Level C        Ȼ (TurboIMAGE)
Global Buffer Length (words) 952  No Logid
Number of Sets              84  Database Created  MON,DEC  9,1991
Number of Items            353  Database has never been DBSTORED
Maximum Buffspecs Set       19  Potential Blockmax (words)  1536
Minimum Buffspecs Set       10  Minimum Buffspecs Required    31
   RECOMMEND Ȼ  Increase Minimum Buffspecs to   31


----------------------------------------------------------------

1) Set (Manual)   BATCH-MASTER:
Search Item Ȼ BATCH-NO            Type Z6
Number of Items              9  Entry Length (words)        14
Number of Paths              1  Blocking Factor             20
Capacity                   201  Block Length (words)       502
Number of Entries          168  Total Number of Blocks      11
Percentage Full          83.6% Clustering Factor          .04
Percentage Secondaries   30.4% Average Secondary Chain    1.44
Percentage Inefficient    1.8% Ldev Num of First Extent     3
Key Value Randomity      .149
RECOMMEND Ȼ  Increase BLOCKING FACTOR to 37 to optimize buffer usage

2) Set (Automatic) OFFICE-MAST:
Search Item Ȼ ORIGINAL-OFF       Type Z4
Number of Items              1  Entry Length (words)         2
Number of Paths              6  Blocking Factor             13
Capacity                    13  Block Length (words)       560
Number of Entries            3  Total Number of Blocks       1
Percentage Full          23.1% Clustering Factor          .17
Percentage Secondaries   33.3% Average Secondary Chain    1.50
Percentage Inefficient     .0% Ldev Num of First Extent     3
Key Value Randomity      .676
RECOMMEND Ȼ  No action at this time

3) Set (Automatic) STOCK-AUTO:
Search Item Ȼ STO-GRAD-PACK       Type Z10
Number of Items              1  Entry Length (words)         5
Number of Paths              1  Blocking Factor             31
Capacity                   201  Block Length (words)       498
Number of Entries          193  Total Number of Blocks       7
Percentage Full          96.0% Clustering Factor          .16
Percentage Secondaries   32.6% Average Secondary Chain    1.48
Percentage Inefficient    3.1% Ldev Num of First Extent     3
Key Value Randomity      .078
RECOMMEND Ȼ  Increase CAPACITY to at least       277
           Percentage full was 96%
RECOMMEND Ȼ  Increase BLOCKING FACTOR to  59 to optimize buffer usage
```

```
DATABASE STATISTICS AND DIAGNOSIS FOR EXBASE.DAVE.PROGDEV     Page:2

14) Set (Detail)   DEALS:
Primary Search Item C  CUS-ACCOUNT      Type X8
Number of Items             54   Entry Length (words)       168
Number of Paths              3   Blocking Factor              3
Capacity                   303   Block Length (words)       541
Number of Entries          212   Total Number of Blocks     101
Percentage Full           70.0%  Number of Deleted Entries   83
Percentage Messy          40.7%  Percentage Deleted        28.1%
% Messy C  16 Sector Cache  26.5% Number of Sorted Paths       0
% Messy C  96 Sector Cache  10.8% Ldev Num of First Extent     2
                                              Chain Lengths
Path  Search-Item    Type    Master-Set    Average    Max    Min
 1    !CUS-ACCOUNT    X8     CUSTOMER        4.7       95      0
 2     DEAL-SUFF      X6     DEAL-MSTR        .9        1      0
 3     DEAL-NO        X6     GLOBAL-MSTR     1.0        6      0
RECOMMEND C  REPACK set along Primary Path (41% messy)
RECOMMEND C  Increase BLOCKING FACTOR to   5 to optimize buffer usage

15) Set (Detail)    MOVEMENTS:
Primary Search Item C  CONTRACT-NO    Type Z10
Number of Items             17   Entry Length (words)       126
Number of Paths              3   Blocking Factor              3
Capacity                   300   Block Length (words)       415
Number of Entries           29   Total Number of Blocks     100
Percentage Full            9.7%  Number of Deleted Entries    0
Percentage Messy          40.0%  Percentage Deleted         .0%
% Messy C  16 Sector Cache  40.0% Number of Sorted Paths       0
% Messy C  96 Sector Cache   .0% Ldev Num of First Extent     4
                                              Chain Lengths
Path  Search-Item    Type    Master-Set    Average    Max    Min
 1    !ST-CONTRACT-NO Z10    CONTRACT-MAST   1.2        4      1
 2     STO-GRAD-PACK  Z10    KEY-MAST       29.0       29     29
RECOMMEND C  Use Serial Access and remove path on STO-GRAD-PACK
 3     ST-UNIT-NO     Z10    NO-MAST         9.7       27      1
RECOMMEND C  Change Primary Path from ST-CONTRACT-NO to STO-UNIT-NO
RECOMMEND C  Increase BLOCKING FACTOR to   6 to optimize buffer usage
RECOMMEND C  REPACK set along Primary Path (40% messy)

--------------------- DIAGNOSIS  COMPLETE  ---------------------
                             (WED, JAN 22, 1992,  5:02 PM)

Further databases....(or press return to exit)....
Database name >>
```

Global Statistics:

The first section of the report gives some global data on the database. The program attempts to access the database directly without going through IMAGE as this gives faster and fuller results. However, if such access is not possible (perhaps due to concurrent access by other users) then the program uses DBINFO calls which restricts the data available.

IMAGE Level

   This indicates which version of IMAGE the database was created with. IMAGE Levels A or B are IMAGE, and level C is TurboIMAGE.

Global Buffer Length (words)

   The length of the IMAGE buffers in the Database Control Block, in words. It is equal to the block size of the set with the largest block in the database. As all sets use the same size buffers it is recommended to modify the blocking factors so as to maximize each set's block size within the limit of the Global Buffer Length and thus reduce I/O (see Potential Blockmax below).

Number of Sets

> The number of sets in the database, or the number of sets accessible via the password level supplied (if DBINFO access method is used).

Number of Items

> The number of items in the database, or the number of items which can be accessed by DBINFO using the password supplied.

Maximum Buffspecs Set

> The largest number of buffers as specified in the database Buffspecs. This value is shown here so that it can be compared to the Minimum Buffspecs Set and Required (see below). Guidelines for this value are difficult to make without detailed knowledge of the use of the database. Experimentation is recommended starting with perhaps twice the Minimum required. On MPE/iX the TurboIMAGE algorithm used means that the exact numbers of buffers set is less relevant than on MPE/V.

Minimum Buffspecs Set

> The smallest number of buffers as specified in the database Buffspecs. If this value is smaller than the Minimum Buffspecs Required (see below) then during some DBPUTs/DBDELETEs and DBUPDATEs, buffers will have to be flushed and re-used before the call can complete, causing a reduction in performance on MPE/V.

Logid

> The name of the LOGID set in the database, if any. This is not available through DBINFO.

Database Created

> The date on which the database was created, or restored from tape. This is not available through DBINFO.

Last DBSTORED

> The date the database was last DBSTORED. This is not available through DBINFO.

Potential Blockmax (words)

> (TurboIMAGE only) This is the potential size that all blocks could be extended to if the full space available for Turbo's buffers were used. The buffers are stored in one Extra Data Segment per database called the DBB which has a maximum size of 32K words. Some of this is reserved for various Turbo Tables, with the rest used for buffers. Taking the Maximum Buffspecs Set (see above) as the number of buffers, the potential maximum size of the buffers can be calculated. This is the figure stated here. In most situations the increase in memory usage due to the increased DBB size can be accomplished without undue memory problems, and the larger buffer sizes should increase throughput. However, as this figure is based on the Maximum Buffspecs Set it may be that more buffers would increase performance C and this would reduce the Potential Blockmax. User evaluation is recommended.

Minimum Buffspecs Required

> When performing a DBPUT on the most complex set in the database IMAGE requires a certain number of buffers to complete all the operations needed. This value can be calculated (see IMAGE manual) and is displayed here. If the number of buffers is less than this, IMAGE must flush some of its buffers so as to re-use them. See Minimum Buffspecs Set above.

Privileged Information Not Available

> This message is displayed if FLEXIBASE has tried to open the database root file directly, but has been prevented for some reason (usually because another user has the database open). The program defaults to using DBINFO to establish the statistics, but cannot access the Logid, Creation Date or the DBSTORE date. Also, for Master Sets, the Percentage Inefficient cannot be established as sufficient data is not available using IMAGE directly. The number of items in the database, when inspected using DBINFO, is the number of "referenced" items only C not the total number of items in the item list.

Third-Party Indexing is enabled

> MPE/iX includes a facility for Third-Party Indexing, whereby proprietary Third-Party indexing products can be added to a TurboIMAGE database. When enabled this will create indexing datasets or files (depending upon the product installed). This message indicates that Third-Party Indexing has been enabled for this database. (See the TurboIMAGE Manual for details of this facility.

TPI Product Name

> This line identifies the name of the Third-Party Indexing product installed.

Number of Index Sets

> The number of datasets, in addition to those specified in the database schema, which are used for Third-Party Indexing. These sets are not visible through QUERY or DBINFO call modes 100 to 400.

Number of Index Files

> The number of external indexing files which are used for Third-Party Indexing. These files have negative file codes and have file names of the form "DBName0A", and "DBName0B" etc.

There now follows the individual set statistics, with the set number, name and type (see the IMAGE/QUERY HP Manual for definitions of standard IMAGE terminology):

> n) Set (Automatic) SET-NAME:, or
> n) Set (Manual)   SET-NAME:, or
> n) Set (Detail)   SET-NAME:

Search Item

> The name of the Search Item for a Master Set.

Primary Search Item

> The name of the Primary Search Item for a Detail Set.

Type

> The item type of the Search Item, e.g. X4.

Number of Items

> The number of items in the set entry.

Number of Paths

> The number of paths into, or out of, the set.

Capacity

> The maximum number of possible entries in the set.

Number of Entries

> The number of entries read by the program in the set, using a serial read.

Percentage Full

> The number of entries read as a percentage of the Capacity.

Percentage Secondaries

> The number of Secondary entries in a Master Set as a percentage of the number of entries in the set. A high percentage can result in poor performance if associated with a high Percentage Inefficient (see next item).

Percentage Inefficient

For a Master Set, this is the number of Secondaries in the set located in a different block to that of the Primary, as a percentage of the number of entries in the set. This is the prime indicator of inefficiency in the set as it may cause extra disc accesses or memory transfers to read the entries.

Percentage Messy

For a Detail Set, this is the percentage of non-contiguous chained entries along the Primary Path pointing out of the block (i.e. entries in the chain do not point to the next sequential entry in the set and point out of the current block). A perfectly REPACKed detail set would have zero Percentage Messy. The higher this figure is, the longer and more inefficient chained access along the Primary Path becomes.

% Messy – 16 Sector Cache (MPE/V systems only)

The same as Percentage Messy above but measured using a "block" size of 16 Sectors (the smallest Cache Domain size in MPE/V) instead of the set's block size. If MPE/V caching was enabled with a Random Cache Quantum of 16 Sectors, this figure would indicate how many non-contiguous Primary path chain pointers lie outside of the 16 Sector block.

% Messy – 96 Sector Cache (MPE/V systems only)

As above, but for 96 Sector "block" size (the largest standard Cache Domain size on MPE/V). Comparison of these three Messy percentages gives some qualitative indication of the extent of the set's messiness, and what effect Caching may have on the set's inefficiency.

% Messy – 1 memory page (MPE/iX systems)

The same as Percentage Messy above, but based on a 4096 byte memory page. On MPE/iX systems, this is the smallest separately-accessible area of disc. It is therefore more significant, from the point of view of detail dataset inefficiency, than the Percentage Messy figure based on IMAGE block sizes.

% Messy – 6 memory pages (MPE/iX systems)

The same as Percentage Messy above, but based on six 4096 byte memory pages. This figure is included to give some idea of the inefficiency, in practical terms, of a detail dataset on an MPE/iX system. Although one memory page is the smallest separately-accessible area of disc under MPE/iX, its prefetch algorithms in fact make it more likely that several contiguous pages will be in memory simultaneously; the exact number will depend entirely on the overall load on the machine. However, a comparison of the three figures quoted can still give a good qualitative feel for how locally contained a chain is within the dataset.

Entry Length (words)

The length, in words, of the set entry.

Blocking Factor

The number of entries per block in the set.

Block Length (words)

Global Buffer Length to optimize the buffer usage and reduce I/O. It is calculated from the Entry Length, Blocking Factor and Number of Paths.

Total Number of Blocks

The total number of blocks in the set.

Clustering Factor

An indicator of how much the entries in a Master Set cluster together. A high value can indicate poor performance, especially if associated with a high percentage of Secondaries. Range is 0.0 to 1.0, with values over 0.25 considered as high.

Average Secondary Chain

> The average length of the Secondary Chains in a Master Set. A Primary with no Secondaries is considered to have a chain length of one.

Number of Deleted Entries

> For a Detail Set this shows how many entries have been deleted from the set, resulting in "holes." A high number indicates a REPACK would improve performance especially on a serial read of the set. To calculate the High Water Mark (i.e. the historic maximum number of entries since creation of the database, or since the last REPACK) simply add this figure to the number of entries in the set.

Percentage Deleted

> For a Detail Set, this is the Number of Deleted Entries as a percentage of the Maximum Number of Entries that the set has ever contained since the last, if any, REPACK (i.e., the "High Water Mark").

Number of Sorted paths

> The number of paths in a Detail Set which have sort items. Too many into one detail set could drastically affect performance on DBPUT and DBDELETE (i.e. dynamic sets).

Ldev Num of First Extent

> The logical device number of the first extent of the dataset. Datasets can have different extents located on different disc drives (logical devices) and so this quoted number may not reflect the location of the bulk of the set. Because of this no recommendations are made with regard to relocating datasets.

Key Value Randomity

> See Appendix F for a detailed discussion of this statistic. It is a measure of how randomly distributed the entries of a master set are. If the blocking factor of the set is 1, this value cannot be determined. Values range typically from 0.001 (bad) to 0.800 (very good). Any value over 0.25 is acceptable.

Third-Party Indexing Keys for this set

> MPE/iX includes a facility for Third-Party Indexing, whereby proprietary Third-Party indexing products can be added to a TurboIMAGE database. When enabled this will create indexing datasets or files (depending upon the product installed). When installed these indexing files or sets may be associated to simple or compound data items. If items in this set have been so configured DIAGNOSE will list the items in the set involved. (See the TurboIMAGE Manual for details of this facility, which were introduced in MPE/iX 4.0.)

Total Size (Sectors)

> The total disc space taken up by the dataset expressed in sectors (4096 Sectors = 1Mb). IMAGE builds the dataset to the size specified by the capacity, even if no entries are added to the set. This value represents the total amount of disc space consumed by the dataset.

Set Used (Sectors)

> The approximate disc space used by the actual entries in the dataset expressed in Sectors. This value is calculated by multiplying the total disc space ("Set Size" above) by the number of entries and dividing by the capacity. In fact there could be "dead" space in the set at the end of each block but this is not accounted for in the calculation. This figure subtracted from the total disc space gives a rough estimate of the amount of "wasted" disc space taken up by the dataset. Every set has to have room for expansion, but leaving too much room could consume unneccessary disc space (see also Capacity Recommendations).

Highwatermark

> The highest record number which has ever contained a data entry in this set. For detail sets this will be equal to the number of entries plus the number of deleted entries. This represents the largest number of entries which this set has ever contained, since the last Repack. A Repack will reduce this number to equal the number of entries.

Weekly Entry Increase

If the Trends database FXSTAT.PUB.HQ exists, and there are sufficient Diagnose entries to
establish a trend, this value will be computed. It represents the average percentage increase (or
decrease) in the number of data entries in the dataset for a week. This is calculated using a "Least
Squares" data fit.

Weekly Entry Growth

If the Trends database FXSTAT.PUB.HQ exists, and there are sufficient Diagnose entries to
establish a trend, this value will be computed. It represents the average absolute increase (or
decrease) in the number of data entries in the dataset for a week. This is calculated using a "Least
Squares" data fit. When Diagnose calculates a new capacity recommendation it will use this value
to ensure that there is sufficient free space to allow for 60 days' worth of entries (or the value
contained in the FXUTIL configuration of Number of Days for Capacity Increases if different).

Capacity Expansion:

For MPE/iX 5.0 or above this indicates that this Detail Set has been enabled for Automatic
Capacity Expansion. This line will be followed by the Maximum allowed Capacity and the
incremental number of entries.

Maximum Capacity

If this Detail Set has been allowed Automatic Capacity Expansion this value represents the
maximum capacity to which the automatic procedure will expand the set.

Incremental Capacity

If this Detail Set has been allowed Automatic Capacity Expansion this value represents the
increase in capacity by which the automatic procedure will expand the set (in number of entries).

B-Trees enabled on this database

If a database has a B-Tree index attached to it, Diagnose will show the above in the Global
Statistics information section.

Set has B-Tree Index keyed on   ALPHA-CODE

(or whatever data item name is used in that set as the key item). Where a master has a B-Tree
index the above line will be displayed.

3.4 RECOMMENDED ACTIONS

After each set's statistics are produced, the DIAGNOSE function intelligently examines them using a rule-
based system to diagnose any potential performance or loading inefficiencies. Some structural and most
chain damage can also be detected and reported on. By expertly correlating the dataset statistics the
DIAGNOSE function can propose recommendations for improved performance, or avoidance of
application failures (by premature filling of datasets for example).

The DIAGNOSE function takes into account the particular version of MPE when formulating
recommendations. The recommendations therefore vary across different versions of MPE.

The following list shows the possible recommendations, but it should be remembered that they are
recommendations only and the database designer/administrator may have specific reasons for not
following the recommendations C thus the DIAGNOSE function output should be evaluated against the
real-life design and environment of your databases and applications.

RECOMMEND C No action at this time

DIAGNOSE/3000 has not detected any inefficiencies and has no recommendations for
improvements.

RECOMMEND C   Increase Minimum Buffspecs to "n"

RECOMMEND C   Increase BLOCKING FACTOR to "n" to optimize buffer usage
The Block Length of the dataset, as determined by the Blocking Factor, is below the Global Buffer Length. Prior to MPE/iX 3.0 TurboIMAGE used fixed size buffers so each block read of a dataset is moved into a buffer which is as long as the Global Buffer Length. Performance can be improved by increasing the dataset's Block Length near to the Global length. This is done by increasing the Blocking Factor in the schema, e.g.: CAPACITY: 1000(12); will tell DBSCHEMA to create a set of Capacity 1000 with a Blocking Factor of 12. Then use the TRANSMUTE option to action the change. Alternatively, use the OPTIMIZE BLOCKING FACTOR function of TRANSMUTE.

RECOMMEND C   Decrease BLOCKING FACTOR to "n" to cater for reduced blockmax
The Block Length of the dataset, as determined by the Blocking Factor, is greater than the recommended BLOCKMAX. When changing the Blockmax this set's blocking factor will need to be decreased for DBSCHEMA to work correctly. This is done by decreasing the Blocking Factor in the schema, e.g.: CAPACITY: 1000(12); will tell DBSCHEMA to create a set of Capacity 1000 with a Blocking Factor of 12. Then use the TRANSMUTE option to action the change. Alternatively, use the OPTIMIZE BLOCKING FACTOR function of TRANSMUTE.

RECOMMEND C   REPACK set along Primary path (x% messy)
A Detail Set with a high Percentage Messy, or a large Percentage Deleted, may be optimized by using option 5 of the FLEXIBASE menu (REPACK) which repacks the set along its primary path. The current percentage messy is displayed so that the user can judge the urgency and potential benefits of this recommendation. On MPE/V systems, this recommendation is based on the Percentage Messy indicated in terms of IMAGE blocks. On MPE/iX systems, the recommendation is based on the messiness over one-page blocks, since this is the smallest area of disc that can be separately accessed.

RECOMMEND C   Increase CAPACITY to at least "n"
     C   Percentage full was x%
Any set with too high a Percentage Full will cause a recommendation to be made which suggests increasing the capacity to provide a more appropriate filling. Use TRANSMUTE or CAPACITY-CHANGE options to effect such a change. See Appendix F for a detailed explanation of the logic used in the capacity recommendations. The second line will be displayed indicating the current percentage full, so that the urgency of the recommendation can be judged.

RECOMMEND C   To reduce Secondary Inefficiency change CAPACITY to "n"
     C   Secondary %age was   x% - will become y% (Ineff<= z%)
     C   Percentage full was a%
A Master Set with too many Secondaries and a high inefficiency can be optimized by altering the Capacity. See Appendix F for a detailed explanation of the logic used in the capacity recommendations. Use TRANSMUTE or CAPACITY-CHANGE options to effect the change. Messages will be displayed indicating the effect of this change on the percentage of secondaries and the percentage inefficient. The second and third line indicate the current secondary percentage and the estimated new percentage for the recommended capacity. The figure in brackets is an estimation of the percentage inefficient for the new capacity (the actual figure should be less than or equal to the figure quoted).

RECOMMEND C   Reduce search item to R2 or change to X, P, Z or U format
A Master Set with a search item of type E or R with length greater than two will produce very high Secondaries due to IMAGE's Numeric algorithm. The solution is to reduce the length, or change to a type which allows IMAGE's Hashing algorithm to optimize the set. Use the TRANSMUTE function to accomplish this (application program changes may also be required).

RECOMMEND C   Change type of search item to X, P, Z or U
As above.

RECOMMEND ₵ Change Primary Path for this dataset
>   In a Detail Set with more than one path it can happen that the wrong path has been selected as the primary path. For example, a path with a maximum chain length of 1 should not be the primary path as REPACK cannot then improve performance by loading chains contiguously. This can be changed using the TRANSMUTE function.

RECOMMEND ₵ REPACK or REPAIR of associated Detail Sets
>   Occasionally an Automatic Master can end up with a different number of entries than its details would imply. This can be repaired by using either the REPAIR or REPACK options, with REPACK optimizing the primary path loading as well.

RECOMMEND ₵ REPAIR of Detail Set
>   DIAGNOSE/3000 has detected an error which can be repaired by using the REPAIR option of FLEXIBASE. For example, a faulty chain pointer in a Detail Set will be detected by using a hash total of forward and backward pointers.

RECOMMEND ₵ Review the need for a sorted chain on path "n"
>   If DIAGNOSE/3000 detects a sorted path which seems unnecessary it will suggest that the need for the path be examined. If it is felt that the sort field is not needed it can be removed using the TRANSMUTE function.

RECOMMEND ₵ Use Serial Access and remove path on ITEM-NAME
>   A Detail Set with an unsorted path whose length is the same as the number of entries in the set can be accessed just as effectively by using Serial reads rather than chained reads. Removing the path will remove some unnecessary overheads, and can be done using the TRANSMUTE option.

RECOMMEND ₵ For MPE/XL 3.0 or above change BLOCKING FACTOR to "n" to give disc space/throughput improvement of nn%
>   From MPE/iX 3.0 the buffer management procedures used by TurboIMAGE were changed. From this point on it will be more beneficial to performance to allocate a Blocking Factor which will minimize the amount of wasted disc space within each TurboIMAGE Block. This is done by changing the Blocking Factor via the TRANSMUTE function. The percentage shown indicates both the saving in disc space, and the I/O performance improvement expected (though the latter will not be so great for random reads on large sets).

3.5 DETECTED ERRORS

The following error conditions may also be reported by the DIAGNOSE module:

Unexpected IMAGE Error ₵ run DBCHECK on this dataset for more details
>   If DIAGNOSE/3000 detects an unexpected internal IMAGE error and cannot recognize its cause (e.g., structural root file damage) then it will recommend using the program DBCHECK (for IMAGE) or TDBCHECK (for Turbo), which come with the FLEXIBASE Installation Tape, to further diagnose the problem. This kind of problem should be extremely rare and is indicative of severe structural damage. It may be necessary to do a full unload/load of the database to correct the problem.

*** Error ₵ entry count on path doesn't match serial read count of "n"
>   This condition indicates some form of structural damage. If there is the possibility of repair then DIAGNOSE will recommend the appropriate course of action. Note that DIAGNOSE checks the count of entries read in a dataset with the figure returned by DBINFO and the count of the master set chain lengths (if this is a detail set).

*** Error ₵ Broken Chain condition detected in this set
>   DIAGNOSE uses a hash total of all forward and backward pointers in a Detail set to detect broken chains. This scheme will detect most broken chain conditions, but not all (corrupted keys or compensating errors for instance). If detected DIAGNOSE will recommend a REPAIR of the set. This condition can only be detected if exclusive access to the database has been obtained.

:FILE Equation not allowed in this option
>   DIAGNOSE will not function correctly if a file equation has been issued on the database. RESET the file equation and use the actual database name.

Unable to open database C :FILE equation not supported
>   As above.

Hashing Error gives nnnn . database xxxx.
>   DIAGNOSE calculates the Primary Set Location of each entry in a Master Set. Each value calculated is checked against the actual location.

Capacity recommendation suppressed
>   A JCW can be set which tells DIAGNOSE not to recommend a new capacity for any set (this can save some time on large databases). It is not recommended for normal use. To implement, enter the MPE command SETJCW FXDIAG02=1. If DIAGNOSE would normally recommend a capacity change and this JCW has been set, the above message appears so as to indicate that a capacity should be reviewed.

Unrecoverable error encountered in Job Mode
>   An error has occurred in a job file when processing an option. When running in interactive (session) mode the user is simply reprompted for a valid entry. However, when running in job mode the FLEXIBASE software has no option but to terminate.

3.6 DIAGNOSE STATISTICS FROM ANALYZE

The DIAGNOSE module will not only compute statistics from the current, static condition of a database but also from the dynamic information stored in a database history file created by the ANALYZE module from Transaction Logging files. At first release these dynamic statistics are quoted but not used in the recommendations of DIAGNOSE. Future releases will extend and refine these figures to provide appropriate performance recommendations.

Examples of these statistics are shown below:

```
<><> DIAGNOSE/3000 <><>   Version D.00.01    (c) Sector 7 Software 1987
                   sold under license by PROACTIVE SYSTEMS worldwide
                                       (WED, SEP 30, 1992, 11:27 PM)

DATABASE STATISTICS AND DIAGNOSIS FOR SALES.DAVE.PROGRAM         Page: 1

Global Statistics:
 IMAGE Level C       - (TurboIMAGE)
 Global Buffer Length (words)   507 Logid                       DSLOG
 Number of Sets                   8 Database Created  SAT, SEP 23, 1989
 Number of Items                 35 Database has never been DBSTORED
 Maximum Buffspecs Set           17 Potential Blockmax (words)   1664
 Minimum Buffspecs Set            8 Minimum Buffspecs Required      8
 Maximum Write Users              2 Num Days Log History          300
 -------------------------------------------------------------------


 1) Set (Manual)    CUSTOMER:
 Search Item - ALPHA-CODE        Type X10
 Number of Items                19 Entry Length (words)          137
 Number of Paths                 8 Blocking Factor                 3
 Capacity                      797 Block Length (words)          571
 Number of Entries             463 Total Number of Blocks        266
 Percentage Full             58.1% Clustering Factor             .00
 Percentage Secondaries      23.8% Average Secondary Chain      1.31
 Percentage Write Access      8.2% Min Free Entries Required       1
 Percentage Inefficient      11.2% Ldev Num of First Extent        3
 Key Value Randomity          .375
    RECOMMEND - No action at this time
```

There are currently 4 statistics which are obtained from a Database History file (a file named "dbnameFX," residing in the same group as the database which is created by the ANALYZE function):

Maximum Write Users

>   This statistic represents the largest number of concurrent write users for this database at any one time (from the logfiles which have had their contents examined with ANALYZE). The users which are logged by Transaction Logging are only those which open the database for write access (DBOPEN modes 1 to 4). Those users who open the database with read-only access are not logged. Thus, this figure represents a minimum number of concurrent users. To this figure should be added the average number of read-only users. A typical multiplier may be 1.5, i.e. two-thirds write users and one third read-only. This multiplier, of course, will vary from installation to installation (and from database to database). When estimating buffer and block sizes this figure becomes important.

Num Days Log History

>   When the Statistics option of the ANALYZE module is run by the Creator of the database, some statistics are stored in a History file called "dbnameFX." Each time a new log file is read, and its Statistics added to the History file, the number of elapsed days which the log file represents is added to the total. Thus, the accumulated Statistics represent so many elapsed days of database access. To be representative at least 30 days of Statistics should have been gathered (i.e., a typical month).

>   The recommended procedure is to run ANALYZE after the logging process has been stopped but before a new logfile is started (i.e., as part of the cycling process of the database logfiles). In this way the full statistical contents of each and every log file will be stored in the database history file.

Percentage Write Access

>   ANALYZE records the number of individual database calls to each set from the logfiles. A log file, however, only logs write access (DBPUT, DBUPDATE or DBDELETE), and not read access. Thus, a set may have zero write access but a large amount of read access (i.e., DBGETs). However, much can be implied from the log files. For every set the total number of DBPUTs, DBUPDATEs and DBDELETEs are computed. Each DBDELETE and DBUPDATE must have been preceded by at least one DBGET, thus the minimum number of DB Calls to each set would be the number of DBPUTs plus twice the number of DBUPDATEs and DBDELETEs. The total of this figure for each set is expressed here as a percentage of the write access to the whole database. In this way one can see which sets are the most dynamic and get the most write access. Optimization can be accomplished by placing the most heavily-used sets on separate logical devices using DBUTIL's MOVE command. However, because read access is not logged this must be taken into consideration. It should be remembered that with MPE V disc caching, write access may well be more of a bottleneck than read access.

Min Free Entries Required

>   When adding or deleting entries to a dataset, free space in the dataset is consumed or released. In a typical month there may well be an extreme program which performs a large number DBPUTs. The Capacity of a dataset must at least be sufficient to accept all of these new entries without filling up. Thus, ANALYZE computes the maximum increase in the number of entries for each set (as being the largest difference between the number of DBPUTs and DBDELETEs for each logfile). The maximum observed figure is stored and reported here. Thus, when estimating a Capacity it must be at least this many entries greater than the current entry count.

>   DIAGNOSE does not currently use this figure in estimating Capacities automatically, but it is planned for a future release.

SECTION 4 ℂ CHECK FUNCTION

4.1 PURPOSE AND OVERVIEW

The purpose of this module is to provide checking of logical relationships within the data, which are implied by the users application design but which are not ensured or enforced by the structural design of the database.

For example, you might have a master set containing CUSTOMER information and another master set containing DELIVERY-ADDRESS information. You may want to be sure that for every customer there is a corresponding delivery address. Thus a periodic integrity check run would make sure that the application was maintaining the relationship between the two, and pick-up any deficiencies caused by program failures or injudicious use of QUERY.

It might be argued that such relationships should be enforced by the design of the database i.e. by inserting paths or consolidating sets, but in the real world, there may be valid practical reasons why the user has a database design which does not precisely match the logical requirement. Indeed the existence of this FLEXIBASE function may allow users to remove certain paths or make other structural simplifications which would improve performance. Furthermore there are some logical relationships (such as a detail-chain total matching a chain-head item value) which cannot be represented or enforced within the IMAGE database structure.

Thus this function can improve the reliability of processing and save time spent on resolving discrepancies by routinely identifying potential problem areas in the data. It will substitute for ad-hoc checking programs which a user might otherwise have to write, and can serve as a debugging aid on user programs which are required to maintain such logical relationships. It can also have uses in identifying unusual or exceptional situations requiring management attention.

Requirements & Pre-conditions:
- The user must supply a database password
- The user must have read, write and save access to the logon group.

4.2 USER DIALOG

The user dialog for the CHECK function is as follows.

```
:RUN FLEXIBAS.PUB.HQ
```

Select Option 3 from the main menu.

```
Enter the name of the main database >>
```

Enter the name of the database within which the integrity-checking is to take place. (This does not exclude rule 1 from doing correspondence checks into other databases which are specified during input for rule 1).

The software does NOT require exclusive access to the database. The user does NOT have to be the creator of the database.

```
Enter database password >>
```

Enter a database password which gives read access to at least those parts of the database involved in the checking. Remember that database passwords are case sensitive.

The complete set of rules is then displayed for the user to select from. Each rule tests a particular logical condition. See Section 4.3 for further information on each rule. A brief list is as follows.

RULE 1: RELATIONAL CORRESPONDENCE

> For each entry in set(A), using the value of item(X) as a search-item, check that there is a corresponding entry in master-set(B).

RULE 2: VALUE OF DETAIL-CHAIN TOTAL MATCHES CHAIN-HEAD

> For each entry in master-set(A), check that the total of items(X+Y..) is equal to the total of items(P+Q..) in all the entries of its chain into detail-set(B).

RULE 3: DETAIL-CHAIN LENGTHS

> For each entry in master set(A), check that the length of its detail chain into set(B) is (=><) length (n).

RULE 4: UNIQUE DATA-ITEM VALUES

> For each entry in set(A), check that the value of data-item(X) is unique.

RULE 5: PARALLEL DETAIL CHAINS

> For each entry in master set(A), check that the length of its detail chain into set(B) is (=><) the length of its detail chain into set(C).

RULE 6: SUBITEMS TOTAL

> For each entry in set(A), check that the total of all subitems in data-item(X) is equal to the value of data-item(Y).

```
SELECT RULE NUMBER (1-6) [or press return to end selections] >>
```

Either select a rule by entering a number in the range 1-6 or press return which will end rule definitions and proceed to the checking phase.

See the appropriate rule in section 4.3 for specific input requirements.

```
Enter limit of Exception-reporting (return=all) >>
```

This limit is provided to avoid situations where thousands of exceptions to the rule might be reported, wasting processing time and paper. To allow for user misunderstandings or erroneous rule specifications, you are advised to set a low limit when first running a rule. Thus, if the user does not expect more than a handful of exceptions to be found, then a limit of 30 might be entered. If a few dozen exceptions are expected, a limit of 100 might be entered. Just pressing return defaults to no limit, but this is not recommended. If the software encounters as many exceptions as the limit, then that fact is recorded on the exception report, and no further testing is done for that application of that rule. Other applications of that rule, and indeed other rules continue to be tested.

```
Confirm (Y/N) >>
```

At this time the selected rule is re-displayed for user examination, and the user can confirm the rule for processing by entering "Y". Alternatively entering "N" will erase the rule and allow selection of another rule.

After all rule definitions have been made, the user presses return at "SELECT RULE NUMBER" and processing starts (unless the run is creating a job file in which case the program will end with a reminder of the job filename).

```
CUSTOMERS        12769 entries to check....    100% Complete
AGENTS             146 entries to check....    100% Complete
TRANSACTIONS    143607 entries to check....     43% Complete
```

The software will process each set in turn if it has been selected as set(A) in one or more rules, and the percentage complete will be displayed in session mode. (In job mode the progress can be examined in the normal way by examining the text file FXPROGRS).

Once processing is complete, an exception report is displayed and printed for each application of each rule. The substituted rule is shown followed by a QUERY style tabulation of the data in every entry which has failed that rule, together with totals or chain-lengths where appropriate. A count of exceptions is also shown. Here is an example:

```
RULE 1: RELATIONAL CORRESPONDENCE

For each entry in set(EMPLOYEES), using the value of data-item(CLOCK-NO) as
search-item, check that there is a corresponding entry in master-set(PERSONNEL).

Exceptions found are :

CLOCK-NO         =PR003
NAME             =JONES M.F.
SALARY           =47500

CLOCK-NO         =FM014
NAME             =SMYTHE A.
SALARY           =35000

            2 exceptions found as detailed above
```

4.3 THE RULES AND THEIR USER INPUT

The CHECK function provides 6 different generic rules concerning data relationships. The rules are presented such that the user can substitute into the text of the rule the specific sets, items, conditions or numbers, which are necessary to describe exactly the checking required. Thus the same rule can be used with different specifics to perform checks on several sets or items within the same database. Indeed each of the 6 rules can be applied in up to 10 different ways within any single program run. The software organises the most efficient method of disc I/O, for achieving the checks requested. Pressing Return in response to any prompt will cause the software to return to the previous prompt. At this point it is appropriate to look at each rule with an example of its use:

RULE 1: RELATIONAL CORRESPONDENCE

For each entry in set(A), using the value of item(X) as search-item, check that there is a corresponding entry in master set(B).

An example of the use of this rule might be as stated in the overview – if the user substitutes for (A), (X) and (B) the names CUSTOMERS, ACCOUNT-NO and DELIVERY-ADDRESS, then the rule reads:

For each entry in set(CUSTOMERS), using the value of item(ACCOUNT-NO) as search-item, there must be a corresponding entry in master-set(DELIVERY- ADDRESS).

So applying this rule would check that for every customer there is in existence a delivery-address, and would report any customers for which there is not a delivery-address.

The rule can also check for relational correspondence to a set in a different database. So if in the STATS database there should be an entry for each CUSTOMER in the CUSTOMER-ANAL set then the rule should read:

For each entry in set(CUSTOMERS), using the value of item(ACCOUNT-NO) as search-item, there must be a corresponding entry in master-set(STATS:CUSTOMER-ANAL).

The search-item, where necessary, can be made up from concatenated pieces of one or more data-items. For example if the 5th and 6th characters of the account-no are the market-code, then a check could be made that an entry for that market does exist:

For each entry in set(CUSTOMERS), using the value of item(ACCOUNT-NO[5,2]) as search-item, there must be a corresponding entry in master-set(STATS:CUSTOMER-ANAL).

It is also possible to reverse the reporting logic on this rule such that the entries reported are those where a corresponding entry IS FOUND in set(B) rather than is missing from set(B). This enables finding which keys are present in both sets. The user can force this mode of reporting by entering a minus sign before the first character of set(B). e.g.   -STATS:MARKETS

This is the dialog between the software and the user for rule 1:

```
Enter name of set(A) >>
```

Enter the name of a detail, automatic or manual set in the main database.

```
Enter name of data-item(X) >>
```

Enter the specification of data to form the search-item into set(B). This can comprise pieces or all of 1 to 6 data-items from set(A) using the syntax

```
        ITEMNAME[StartChar,CharLength]+ ....
```

The square brackets and contents are omitted if the whole item is to be used.
Evidently, the total length of the item(s) or pieces of item(s) comprising item(X) must not be longer than the length of the search-item of master-set(B) - this will result in an error message. The length is, however, permitted to be less, in which case the item(X) will be padded with spaces to the search-item length (a warning is issued to the user).

```
Enter name of master-set(B) >>
```

Enter the name of a manual or automatic set in any database using the syntax:
DBNAME.GROUP.ACCOUNT:SETNAME The Account or Group.account can be omitted if it is the same as the logon. If the set is in the main database then only the SETNAME is required. A database password will be requested for any specified database. (Remember that upper and lower case are critical in database passwords.) This password must provide at least read access to the set specified. MPE security read access to the group.account (if specified) is also necessary unless the database is RELEASED. A remote database can be referenced by use of a file equation.

RULE 2: VALUE OF DETAIL-CHAIN TOTAL MATCHES CHAIN-HEAD

For each entry in master-set(A), check that the total of items(X+Y..) is equal to the total of items(P+Q..) from all the entries of its chain into detail-set(B).

An obvious example for this relationship is where the total of the CREDIT and DEBIT amounts in each of the TRANSACTION detail entries in a customer's chain should add up to the BALANCE as held in the CUSTOMER master entry itself. So the substituted rule might read:

For each entry in master-set(CUSTOMERS), check that the total of items (BALANCE+BALADJUST) is equal to the total of items (DEBIT+CREDIT) from all the entries of its chain into detail-set (TRANSACTIONS).

Because arithmetic is effected after conversion into long real data, in order to avoid floating-point inaccuracies the equality is judged to +/- 0.005 .

This is the dialog between the software and the user for rule 2:

```
Enter name of master set(A) >>
```

Enter the name of any manual master set in the main database.

```
Enter item-names(X+Y..) >>
```

Enter the names of 1 to 6 data-items in master-set(A) joined by plus signs. These items will be added together to form the chain-head total. Minus signs can also be used to indicate subtraction of item values, but the underscore character ("_") is used for minus rather than hyphen - to avoid confusion with hyphens in data-item names. e.g. BALANCE+BALANCE-FWD_OPENING-BAL

```
Enter name of detail set(B) >>
```

Enter the name of any detail set connected by a path to master set(A).

```
Enter item-names(P+Q..) >>
```

Enter the names of 1 to 6 data-items in detail set(B) joined by +/_ signs as above.

RULE 3: DETAIL-CHAIN LENGTHS

For each entry in master set(A), check that the length of its detail chain into set(B) is (=><) length (n).

A common example might be where there should be a one-to-one relationship between an automatic and detail set. In other situations there should always be at least one detail entry chained to each master entry. Alternatively this rule could be used to identify inactive ACCOUNTS where there are no current TRANSACTIONS, or dead ORDERS where there are no outstanding ORDER-LINES. In some circumstances there should be a specific number of entries on each chain - one for each class of product, or maybe there should never be more than a specified number of product classes. Alternatively this option could be used to list exceptional conditions such as more than 20 outstanding INVOICES or less than 2 TRANSACTIONS in a year, or more than one DELIVERY-ADDRESS.

This is the dialog between the software and the user for rule 3:

```
Enter name of master set(A) >>
```

Enter the name of any manual or automatic master set in the main database.

```
Enter name of detail set(B) >>
```

Enter the name of any detail set which is connected by a path to master set(A).

```
Enter the chain length condition "=" or "<" or ">" >>
```

Enter one of the 3 given symbols to indicate the condition to be checked.

```
Enter the length for chain to be compared against >>
```

Enter a number representing the chain-length to be used in the checking.

RULE 4: UNIQUE DATA-ITEM VALUES

For each entry in set(A), check that the value of data-item(X) is unique.

IMAGE enforces the uniqueness of search-item values in a master set, but it might be required to maintain uniqueness of the value of some other item in a detail or master set. This rule can also identify duplicate data-item values or check the viability of potential search-items before a structural change. Since the data to be checked can be a concatenation of pieces of several data-items, complex checks of uniqueness/duplication can be made.

This is the dialog between the software and the user for rule 4:

```
Enter name of set(A) >>
```

Enter the name of any set in the main database.

```
Enter name of data-item(X) >>
```

Enter the specification of data to be checked for uniqueness. This can comprise pieces or all of 1 to 6 data-items from set(A) using the syntax:

```
ITEMNAME[StartChar,CharLength]+ ...
```

The square brackets and contents are omitted for the whole item.
e.g. ACCOUNT-NO[1,2]+INDUSTRY-CODE+AGENT[1,4]

RULE 5: PARALLEL DETAIL CHAINS

For each entry in master set(A), check that the length of its detail chain into set(B) is (=><) the length of its detail chain into set(C).

For example, perhaps each CUSTOMER should have the same number of TRANSACTION entries as SALES-ANALYSIS entries; or where a detail set has had to be split into two sets then the two chain lengths should always be the same. Alternatively this rule could identify where there might be fewer INVOICES than DELIVERIES etc.

This is the dialog between the software and the user for rule 5:

```
Enter name of master set (A) >>
```

Enter the name of any manual or automatic master set in the main database.

```
Enter name of detail set(B) >>
```

Enter the name of any detail set which is connected by a path to master set(A).

```
Enter the chain length condition "=" or "<" or ">"  >>
```

Enter one of the 3 given symbols to indicate the condition to be checked.

```
Enter name of detail set(C) >>
```

Enter the name of any detail set, other than set(B), which is connected by a path to master set(A).

RULE 6: SUBITEMS TOTAL

For each entry in set(A), check that the total of all subitems in data-item(X) is equal to the value of data-item(Y).

For example, for ease of reporting users sometimes hold an annual total in one data-item as well as the individual month values in a sub-item array. This rule will routinely identify where the two are not equal.

Because arithmetic is effected after conversion into long real data, in order to avoid floating-point inaccuracies the equality is judged to +/- 0.005 .

This is the dialog between the software and the user for rule 6:

```
Enter name of set(A) >>
```

Enter the name of any set in the main database.

```
Enter name of data-item(X) >>
```

Enter the name of a data-item in set(A) which contains subitems.

```
Enter name of data-item(Y) >>
```

Enter the name of a data-item in set(A) which does not contain subitems.

4.4 OTHER POINTS

If a CHECK run is aborted for any reason, the exception reports will be left on disc (each rule as a separate text file). The file name for each exception report is "EXRPTraa" where r=Rule and aa=application. So for example the second application of rule 3 would have a report filename of "EXRPT302". These reports can be viewed in an editor and purged. They will in any case be purged by the next checking run using those filenames.

In rules 2,3 and 5 it is assumed that the master and detail set pair is joined only once by a path (other paths to other sets are of course acceptable). In the rare case where other paths exist between the same sets, this software will identify and use the first of the paths declared in the detail part of the schema.

There is a slight possibility that with some rules, other users on the database might coincidentally update the same entries as are being checked, and therefore give false exceptions. If this unlikely event is of concern then the user should arrange for exclusive use of the database.

SECTION 5 ₵ ANALYZE FUNCTION

5.1 PURPOSE AND USER DIALOG

This module reads and analyzes database logfiles created using the Transaction Logging facility of IMAGE and TurboIMAGE. See Appendix G for a discussion of logging and how it can be used for more than just database recovery.

In addition to providing extensive enquiry and statistical facilities for any particular logfile, the module also creates and updates History Files for a database. These files store actual usage figures which provides the DIAGNOSE module with further statistics upon which to make performance recommendations (these figures are currently only reported in DIAGNOSE; their use in performance analysis and recommendations is planned for a future release).

A utility program called FXUTIL is provided to configure the output from the ANALYZE module, in terms of output device, page length/width etc. See Section 24 for more details on this utility.

```
Requirements & Pre-conditions:
The user must have read access to the database.
The user must be able to open the database in mode 5 or 6.
The user must be able to open the logfile for read access.
The user must be the database creator to be able to maintain the History Files.
```

The following is the dialog between the user and the FLEXIBASE software:

```
:RUN FLEXIBAS.PUB.HQ
```

Select option 4 from the menu.

```
Do you want to run Interactively now or create a streamed Job file?
```

As with all FLEXIBASE options, the user may run the program interactively to perform the analysis now, or create a batch job which may be streamed to perform the analysis at a later date.

The following titles will be displayed on the screen:

```
<><> ANALYZE/3000 <><> Version D.00.01        (c) Sector 7 Software 1989
                    sold under license by PROACTIVE SYSTEMS worldwide
                                         (SUN, SEP 17, 1989,  4:21 PM)
Enter Database name >>
```

Enter the name of an existing database which is, or has been in the past, enabled for logging. This name may be fully qualified provided that the user has normal DBOPEN access to the database with regard to MPE security. No matter how many databases log into the specified log file, only transactions relating to this database will be selected. The database need not reside in the same group/account as it was when the logfile was created; however the database name entered is matched against the name of the database recorded in the logfile (i.e., the original name). Thus, if the database has been moved a file equation must be used to ensure the original name is re-directed to the new location (i.e., :FILE oldname=newname).

```
Enter Password >>
```

Enter a valid password for this database. The program will attempt to open the database in mode 5 or 6. Other users may have the database open while the analyzing is in progress. This password should give read access to all sets in the database for which log entries will be selected and examined. Enter creator password of ";" if you wish to have the Database History File updated (you must be the database creator to update the Database History File).

```
Enter Logid Password >>
```

When a database is set up for logging with DBUTIL the creator of the database enters the Logid (see Appendix G). This logid may have a password associated with it for security reasons. If this is the case ANALYZE will ask the user to enter this password. It will not proceed with this database until the correct password has been entered. Press return to select another database.

```
Enter Logfile Name >>
```

The program will display the current Logging Identifier (if any) for this Database. Enter the name of a valid IMAGE or TurboIMAGE log file which relates to this database. You must have read access to this file, and it must have a file code of LOG (1090). The file must relate to the current structure of the specified database exactly (i.e., there should have been no database transformations since the logfile was created). The file may be the current logfile, or older.

If automatic changing of log files has been enabled the first log file in the set should be entered here (i.e., 001 files). ANALYZE will automatically open subsequent logfiles for the statistics option, and for the other options if the end date/time is later than the last entry in the specified log file.

Log files also can be re-directed by file equations, the only proviso is that any continuation logfiles cannot then be automatically opened in sequence as it is their original names which appear in the log file. In this case ANALYZE will ask the user for the new name as needed. Entering "?" will give a list of all logfiles in the logon group.

The program will examine this log file to ensure that it is a valid IMAGE or TurboIMAGE logfile. The number of records in the file, as well as the start and end times of the log records (in this file only) will now be displayed, e.g.:

```
Logfile MSARLOG.DAVE.PROGRAM contains        295 log records
Logfile starts at 22:28 25/08/88
and runs until    17:39 22/06/89
```

A short overview of the options within the ANALYZE module will now appear on the screen:

```
You can analyze the above Database and Logfile in three different ways:
-------------------------------------------------------------------
 A ₵  Audit Trail Option    : A sorted trail of selected log entries
 E ₵  Logfile Enquiry Option : Full details of selected log entries
 S ₵  Statistics Option      : Logfile and historical statistics
-------------------------------------------------------------------
Enter one of the above letters to choose an option,  or press RETURN to
end the analysis of this Database/Logfile combination.

Select Option (A,E or S) >>
```

Enter a single letter (A, E or S) to select one of the above ANALYZE options:

The (A)udit Trail option will provide a listing of selected entries from the log file in the form of one line per log file entry (Put, Update etc.). This listing will be sorted in order of either User Name, Program Name or Logical Device Number. Optionally a listing only of the text contained in DBBEGIN, DBMEMO or DBEND calls may be produced (again sorted). Selected entries may be reported by specifying selection criteria on dataset, user, program, ldev and time.

The (E)nquiry option will detail each log file entry in full detail, with date and time etc., along with each data item (and sub-item) and its value. The log entries may be selected in more detail than for the Audit Trail option, by selecting a specific data item value condition (e.g., AMOUNT > 1000000).

The (S)tatistics option will provide a list of all processes which accessed the database (for the entire logfile). For each process the number of DBPUTs etc. will be displayed, followed by a similar list for each dataset. If you are the database creator then a file will be created/updated called "dbnameFX." This

file will contain more detailed statistics and will become a historical record of the database usage. The DIAGNOSE module will use this file to get a picture of how the database is actually used. If this file exists, its historical statistics will also be displayed.

```
Enter Lotus "Comma-Separated-Variable" File Name >>
```

This question will only appear if Lotus compatible output has been configured for ANALYZE, using FXUTIL (see section 24 for further details).

5.2 AUDIT TRAIL OPTION

The purpose of this option is to produce a concise report of the entries in the log file. This option will produce a one line entry for each database call (DBPUT, DBUPDATE, DBDELETE, DBBEGIN, DBEND or DBMEMO), listing in summary as much data as will fit on one line (given the configured output width).

Log entries which relate to specific sets, or which have been created by particular users, programs or logical devices may be selected for this report. In addition a range of dates and times may be input to further refine the quantity of data produced.

Once the log entries are selected the subsequent report may be sorted by User Name, Program Name or Logical Device Number. Within each sorted report the log entries will appear in chronological order of the transactions appearance in the logfile (by date/time).

The dialog for selection within this option is as follows:

```
Sort by Users, Ldev Nos or Processes   (U,L or P) >>
```

The output from this Audit Trail selection may be sorted in one of three possible ways: (U)ser Name, (P)rogram Name or (L)ogical Device Number. Enter a single letter to select the sorting method.

Print only DBBEGINs, ENDs & MEMOs with Text (Y/N) >>

An audit trail may be generated which only details the text entries of calls to DBBEGIN, DBEND and DBMEMO. This greatly reduces the quantity of output (saving even more trees), and can provide a concise report of the operations done on the database by user, process or logical device. However, to accomplish this the user application programs must have been coded to place relevant text into these calls. Some packaged software may not do this. (See example output below.)

Following this question the user may specify six selection criteria. These are detailed below. In each case entering Carriage Return only will select the default values. These criteria are also validated (e.g. the named dataset must exist) and the user is re-prompted in the case of an error (except in job mode when the program must terminate).

```
Enter Dataset Name  >>
```

Log entries may be selected which relate only to a certain dataset or sets. Enter the name of a valid dataset within this database. The wild card character "@" may be used to select a subset of sets, e.g. "DET@" will select only those log entries relating to sets whose names start with "DET." Entering "?" will give a list of all sets within the database. The default value is "@" ₵ meaning all datasets within the database.

```
Enter User Name      >>
```

Enter a valid user name (may be fully qualified). Only log entries made by users matching this name will be selected. The wild card character "@" is allowed, thus entering "U@.A@" will select only users whose names start with U in accounts which start with A. "@.@," or return, will select all users (the default).

```
Enter Program Name   >>
```

Enter a valid program name (may be fully qualified). Only log entries made by programs matching this name will be selected. The wild card character "@" is allowed, thus entering "P@.G@.A@" will select programs whose names start with P in groups which start with G and accounts which start with A. "@.@.@," or return, will select all programs (the default).

```
Enter Ldev Number    >>
```

Enter a valid logical device number. Only log entries made by users running on this logical device number will be selected. "@," or return, will select all logical device numbers (the default).

```
Enter start time/date >>
```

Enter the time and date for the beginning of this audit trail selection. The date format accepted is that configured by using the program FXUTIL.PUB.HQ. Date format 0 is "hh:mm dd/mm/yy," date format 1 is "hh:mm mm/dd/yy," and date format 2 is "hh:mm ddmmmyy." An invalid date (e.g. 45/12/89) will be rejected. Only log entries on or after this time will be selected. Entering return will default this value to the start time/date of the logfile selected.

```
Enter end   time/date >>
```

Enter the time and date for the end of this audit trail selection. The date format accepted is that configured by using the program FXUTIL.PUB.HQ. Date format 0 is "hh:mm dd/mm/yy," date format 1 is "hh:mm mm/dd/yy," date format 2 is "hh:mm ddmmmyy." Only those log entries on or before this time will be selected. This time may be later than the end time/date of the opened logfile, and should be so in the case of continuation logfiles created with automatic changelogging. This time may not be less than the start time/date. Entering return will default this value to the end time/date of the logfile selected.

```
Proceed with this selection (Y/N) >>
```

Once the selection criteria are successfully entered the program first confirms that the criteria that have been entered are correct. Enter Y to proceed and the program now reads the logfile, or files. Entering N at this point will take the user back to selecting Audit Trail, Enquiry or Stats.

The interactive user will see the times of the individual log records appearing on the screen, indicating the progress of the selection process. In addition the dates and times at which logging was started or restarted are noted. An example screen would be:

```
Reading Log File...................
 22:28 25/08/88 <><>  Logid DSTSTLOG Started    <><>
 18:29 20/04/89 <><>  Logid DSTSTLOG Restarted  <><>
 13:42 04/05/89 <><>  Logid DSTSTLOG Restarted  <><>
 17:20 09/05/89 <><>  Logid DSTSTLOG Restarted  <><>
 17:28 09/05/89 <><>  Logid DSTSTLOG Restarted  <><>
 18:07 11/05/89 <><>  Logid DSTSTLOG Restarted  <><>
 19:07 05/06/89 <><>  Logid DSTSTLOG Restarted  <><>
 17:38 22/06/89 <><>  Logid DSTSTLOG Restarted  <><>
       7 entries selected
```

If continuation logfiles were encountered they will be opened automatically and read up to the last file, or the end time/date criterion is exceeded. In this case the user will also see messages indicating which files were opened:

```
Reading Log File...................
 18:32 30/08/89 <><><>   Logid DSTEST   Started   <><>
 18:32 30/08/89 <> Logfile DS002.DAVE.PROGRAM Opened <>
 18:32 30/08/89 <> Logfile DS003.DAVE.PROGRAM Opened <>
 18:32 30/08/89 <> Logfile DS004.DAVE.PROGRAM Opened <>
 18:32 30/08/89 <> Logfile DS005.DAVE.PROGRAM Opened <>
 18:33 30/08/89 <> Logfile DS006.DAVE.PROGRAM Opened <>

Enter File Name of Continuation Logfile >>
```

The names of continuation logfiles are stored in the logfiles themselves, and so in most cases ANALYZE will be able to automatically open each successive logfile. However, if the files have been renamed or moved to another account/group then the files may not be available to be opened. In this case ANALYZE will ask the question above to get the name of the next logfile to open. Enter the fully qualified name of the next file in the sequence. Entering nothing (press return only) will stop the selection process.

Once the selection process has completed the program notes how many log entries were selected for later reporting. This number will always be lower than the total number of log records in the log file. This is because there are log records for starting and stopping the logging process, opening and closing databases etc. There are also many "null" log records written when logging is stopped. In addition, one database call to DBPUT may span more than one log record. Each log record is only 256 bytes wide, so a put to a set with an entry size larger would result in more than one log record to store the contents of the entry. A log entry is thus one or more log records relating to a single database call.

These selected entries are then sorted and the output produced on the configured device.

5.3 EXPLANATION OF AUDIT TRAIL OUTPUT

An example of the output from the Audit Trail option follows. The formal file designator for the output file is FXAUDIT, and is automatically opened as specified in the configuration file maintained by FXUTIL. The device name may be overridden by a file equation if required. The example below is for a record width of 80 characters, but a maximum width of 256 characters is possible. This example is for an audit trail of all the logfile entries for the CUSTOMER dataset:

```
<><> ANALYZE ₡ Logfile Audit Trail   (Run: SUN, SEP 17, 1989,  4:24 PM)
From 08:28 31/08/88 until 17:39 31/08/88 Database SALESM.DAVE.PROGRAMS
User @.@                                 Process  @.@.@
LDev @                                    Logfile  MSLOG.DAVE.PROGRAMS
Set  CUSTOMER                             Page:    1
----------------------------------------------------------------------

 Time    Ldev Dataset        Key Value    Data Entry Values

User PROGS.PROGRAMS,DAVE     Process QUERY.PUB.SYS
31/08/88
10:21 Put  71 CUSTOMER       test         ACCOUNT-NO    = 1234
10:22 Upd  71 CUSTOMER       NORD         CREDIT-LIMIT  =       966.00
10:22 Upd  71 CUSTOMER       NORD         ADR-LINE-3    = East Germany
12:07 Upd  71 CUSTOMER       test         ACCOUNT-NO    = 9999
12:08 Upd  71 CUSTOMER       test         TYPE          = AB
13:47 Upd  27 CUSTOMER       test         ADR-LINE-3    = THIRD LINE
13:48 Upd  27 CUSTOMER       test         AGENT-CODE    = 12


<><><><><><><><><>      End of Audit Trail Option     <><><><><><><><><>
```

Because this report has been produced on a screen configured at only 80 characters wide each line has only enough room to display one data item in addition to the key value. The wider the report the more items will be displayed. Each item takes up 38 characters for its name and value (truncated if needed). The order of the items will be the order of their appearance in the log records (which in turn will be the order specified in the item lists of the DB calls themselves).

The heading of each page details the selection criteria used (i.e., which set or logical device). The values "@" or "@.@" represent the default values and indicate that all values for this criterion will have been selected. In the above example all Logical Devices, Programs and Users would have been selected.

Definitions for each part of the heading of this report follow:

<><> ANALYZE ₡ Logfile Audit Trail   (Run: SUN, SEP 17, 1989, 4:24 PM)
      The report header detailing the date and time at which this report was run.

From 22:28 25/08/88 until 17:39 22/06/89

The start and end times specified by the user. The default values are taken from the timestamps in the first log record of the file and the last correctly posted log record in the file.

Database SALESM.DAVE.PROGRAMS

The database name for which this report has been produced.

User @.@

The user name selection criterion. In this case @.@ indicates a selection of all user names (the default C i.e., any user of any account).

Process @.@.@

The program/process name selection criterion. In this case @.@.@ indicates a selection of all programs (the default C i.e., any program in any group in any account). Process in this context is synonymous with Program.

LDev @

The logical device number selection criterion. In this case @ indicates a selection of any logical device (the default).

Logfile MSLOG.DAVE.PROGRAMS

The name of the logfile analyzed. In the case of continuation logfiles this will be the name of the LAST logfile examined in the logfile set.

Set CUSTOMER

The dataset selection criterion. In this case only log entries relating to the CUSTOMER set will be selected. The default is @, indicating any set.

Definitions of each part of the body of this report follow:

Time

The time at which this entry in the log occurred. This is not the time at which it was physically written to the logfile because the Transaction Logging process uses buffered I/O. Thus, this is the actual time at which the database call was executed. All times are displayed in 24 Hour clock format, i.e. 10:21 is 10:21 AM and 22:21 is 10:21 PM.

Following the time is a three-character abbreviation representing the type of database call selected by this option. The calls which could be selected by the audit trail option are:

Upd	C	DBUPDATE
Put	C	DBPUT
Del	C	DBDELETE
Beg	C	DBBEGIN
End	C	DBEND
Mem	C	DBMEMO

Ldev

The logical device upon which this database call was made. Each process can run upon only one logical device number. Batch jobs usually will have logical devices of 10 or less (depending on your particular configuration of I/O devices).

Dataset

The dataset name for which the database call has been made.

Key Value

Every DBPUT, DBDELETE or DBUPDATE to a master set logs not only the data which is changing but also the Search item value for that Master Set. In the case of detail sets however this information is only logged for DBPUTs and DBDELETEs (in fact the entire entry is logged for these calls). As a detail set may have up to 16 different Search items in it ANALYZE will display here the value of the Primary Path Search Item only.

However, because no Search Items may be modified by a DBUPDATE no Search Item values are logged for detail set DBUPDATE calls. In this case ANALYZE will attempt to read the dataset at the record number specified in the log entry. However, there is no guarantee that the value so read (if any) represents the same entry that was updated (because subsequent puts and deletes to the set may have overwritten the entry). In this case the Key Value is displayed with a "?" at the end to indicate that this is potentially inaccurate.

Data Entry Values

Depending upon the width of output, there will be displayed a list of the data items logged, in order of their appearance in the logfile entry. For each data item its name will be followed by an equals sign, "=," and the first 20 characters of its value. In the case of DBUPDATEs this represents the newly updated value of the item (the old value is also displayed in the Enquiry option).

User PROGS.PROGRAMS,DAVE

This example report has been sorted into User order. Each new User/Program combination will be printed on a new page, sorted by User. The format is User.Account,Logged-OnGroup.

Process QUERY.PUB.SYS

A process is defined as a particular program running in a specific session. In this context the word Process is synonymous with the word Program. The value displayed here is the MPE file name of the program which made the log entries (by calling the DB intrinsics and opening the database in mode 1 to 4).

Following the User/Process names will be a date in which the log entries were made. This date can be displayed in 3 formats (see above).

Selecting only those logfile entries for calls to DBBEGIN, DBMEMO and DBEND one can produce a very concise trail of work done by individual users. These calls all contain a Text parameter which allows the programmer to enter text into a logfile. Using ANALYZE this text can now be retrieved so as to create a daily trail of update access to a database. Of course, user programs will have to be modified to enable this facility. In this way large transactions involving many datasets can be recorded on an audit trail as a single line C by entering text into either the DBBEGIN or DBEND call.

For example, consider the following selected audit trail produced for a Stock/Order database:

```
<><> ANALYZE C  Logfile Audit Trail    (Run: THU, SEP 21, 1989,  1:59 PM)
From 22:14 19/09/89 until 13:51 21/09/89 Database STORES.WORKS.PLC
User @.@                                 Process  @.@.@
LDev 56                                  Logfile  STLOG.WORKS.PLC
Set  @                                   Page:     1
-------------------------------------------------------------------

 Time    Ldev Text

User ABSTORES.PLC,WORKS        Process STSYSP.WORKS.PLC
20/09/89
08:34 Beg  56 Order Line A11868/ 1 amended/cancelled
08:35 Beg  56 Line  1 of Order A11868 Received
08:35 Beg  56 Line  A of Order A11868 Received
12:24 Beg  56 Line  B of Order A11867 Received
12:45 Beg  56 Cancelled Order A10356
21/09/89
08:43 Beg  56 Line  A of Order A11606 Received
08:43 Beg  56 Line  B of Order A11606 Received
08:45 Beg  56 Invoice Passed for Order Number A11606, Line  A
08:45 Beg  56 Invoice Passed for Order Number A11606, Line  B
```

Text

The value displayed here is the text coded into the DBMEMO, DBBEGIN or DBEND call. In QUERY the commands TRANSMEMO, TRANSBEGIN and TRANSEND perform the same operation. This text may be any user-written note describing the transaction in as much detail as is required to identify the operation being performed. Each operation, or transaction, may actually make many DBPUT, DBUPDATE or DBDELETE calls to many different sets. Thus a one line audit or work trail can be produced in summary form. The details of each transaction can be enquired using the Enquiry option, and specifying a narrow range of date/times to capture only those entries required.

This facility could replace the use of MPE ASCII files as trails of work done by specific programs. The advantage here is that log files are not as easily deleted as plain MPE files, and the logging to the file is performed by the MPE file system and is thus much less susceptible to locking/concurrent access problems.

Reports produced on $STDLIST

If the FXAUDIT output (or any of the other two options) has been directed to $STDLIST, each screenful of output will be followed by a pause until the user presses Return. The program displays the following:

```
Press Return for next Page (E exits) >>
```

This is designed to allow the user time to examine the data displayed on the screen. After the user enters Return the program clears the screen and displays the next page of output. Entering "QUIT", "EXIT", "E", or "ABORT" at this stage terminates the display of the selected entries.

Note: although the output may be re-directed by use of a file equation to $STDLIST the output width will remain as specified in FXUTIL.

5.4 LOG ENQUIRY OPTION

Having identified the log entries one is interested in (perhaps by running the Audit Trail option, or by noting some data problems), one can use this option to produce a full report of everything in each relevant log entry. By definition this will produce a lot of output, and it is not recommended to attempt a log enquiry of every record in a normal log file. (A 10,000 record log file could produce up to 1,500 sheets of output at 66 lines per page!) Thus, the normal use of this option is to select relevant subsets of the logfile. For example, only those entries which change a certain data item, or entries for a certain set by a certain user etc.

Seven selection criteria are available. Enter a valid value for each, or enter return for the default values.

```
Enter Dataset Name   >>
```

Enter the name of a dataset within this database. The wild card character "@" may be used to select a subset of sets, e.g. "DET@" will select only those log entries relating to sets whose names start with "DET." The default is all sets selected. Enter "?" for a list of all sets within the database.

```
Enter Program Name   >>
```

Enter a valid program name (may be fully qualified). Only log entries made by programs matching this name will be selected. The wild card character "@" is allowed, thus entering "P@.G@.A@" will select programs whose names start with P in groups which start with G and accounts which start with A. "@.@.@" (or return) will select all programs (the default).

```
Enter User Name      >>
```

Enter a valid user name (may be fully qualified). Only log entries made by users matching this name will be selected. The wild card character "@" is allowed, thus entering "U@.A@" will select only users

whose names start with U in accounts which start with A. "@.@" (or return) will select all users (the default).

```
Enter Ldev Number    >>
```

Enter a valid logical device number. Only log entries made by users running on this logical device number will be selected. "@" (or return) will select all logical device numbers (the default).

```
Data Item Condition >>
```

This criterion allows the user to select out only those entries relating to a specific data item. In addition a condition upon the allowed range of data item values is specified. Enter a valid data item name, followed by one of the allowed relational operators >, <, =, <>, >= or <=, followed by a value for this data item. Matching entries will have their entire contents output, not just that of the data item concerned.

If a data item has sub-items (compound item) then all sub-items will be checked for matching values. Thus, the user should not specify a particular sub-item.

For example, entering ACCOUNT-NO>1234 will select those entries which contain the field ACCOUNT-NO with a value greater than 1234. Only one condition is allowed at present. Quotation marks around values are allowed but are not mandatory.

For DBPUTs any entry for any set which contains a matching data item with a matching value will be selected. For DBDELETEs any entry for any set which contains a matching data item with a matching value before deletion will be selected. For DBUPDATES any entry for any set which contains a matching data item with a matching value as the newly updated value will be selected. (Note: Detail set key values are not logged for calls to DBUPDATE and thus key values cannot be used to select DBUPDATES.)

```
Enter Start Time/Date >>
```

Enter the time and date for the beginning of this log enquiry selection. See the Audit Trail option for more details.

```
Enter End   Time/Date >>
```

Enter the time and date for the end of this log enquiry selection. See the Audit Trail option for more details.

```
Proceed with this selection (Y/N) >>
```

Having selected the required criteria the program first verifies that the selection criteria are correct. Enter Y and the program will now read the specified log file (and any continuation logfiles) up to the end date/time specified (or end of file). Entering N at this point returns the user to selecting Audit Trail, Enquiry or Stats.

The selected entries are output as they are selected, i.e. they are not sorted but appear as they are read from the logfile, in the order of entry in the file.

When complete the number of log entries selected will be displayed. See the Audit Trail above for an explanation of what this number refers to. For the same selection criteria as the Audit Trail option a different number may well appear here as extra details are displayed in the Enquiry option (DBOPEN and DBCLOSE calls).

5.5 EXPLANATION OF LOGFILE ENQUIRY OUTPUT

An example of the output from the Logfile Enquiry option follows. The formal file designator for this report is FXENQUIR, and is automatically opened as specified in the configuration file maintained by FXUTIL. The device name may be overridden by a file equation if required. The example below is for a

record width of 80 characters, but a maximum width of 256 characters is possible. The full values of each data item are displayed where possible, and are only truncated if insufficient output width is available (as is the case in the example below).

This example is for an enquiry of all the logfile entries made using QUERY.PUB.SYS:

```
<><> ANALYZE C  Logfile Enquiry        (Run: SUN, SEP 17, 1989,  4:29 PM)
From 08:28 30/08/88 until 17:39 31/08/88  Database SALESM.DAVE.PROGRAMS
User @.@                                   Process  QUERY.@.@
LDev @                                      Logfile  MSLOG.DAVE.PROGRAMS
Set  @                                      Item     @=@

-----------------------------------------------------------------------

LOG RECORD NO   =          66      DATABASE NAME   = SALESM.DAVE.PROGR
DATE            =    31/08/88      SECURITY LEVEL  =     64
TIME            = 10:20:43.5       USER ID         =
LOG NUMBER      =          13      DBOPEN MODE     =      3
LDEV NUMBER     =          71
DST/TRANS NO    =  1037/
SET RECORD NO   =
IMAGE CALL      = DBOPEN
DATASET NAME    =
PROGRAM NAME    = QUERY.PUB.SYS
USER NAME       = TEST.PROGRAM,DAVE

LOG RECORD NO   =          67      ALPHA-CODE      = test
DATE            =    31/08/88      ACCOUNT-NO      = 1234
TIME            = 10:21:37.1       NAME            = test account
LOG NUMBER      =          13      ADR-LINE-1      = first line
LDEV NUMBER     =          71      ADR-LINE-2      = second line
DST/TRANS NO    =  1037/   1       ADR-LINE-3      = third line
SET RECORD NO   =         289      CREDIT-LIMIT    =      234.00
IMAGE CALL      = DBPUT            B/F-BALANCE     =        0.00
DATASET NAME (M)= CUSTOMER         CURRENT-BALANCE =       34.00
PROGRAM NAME    = QUERY.PUB.SYS    BAL-1-MONTH     =        2.00
USER NAME       = TEST.PROGRAM,DAVE BAL-2-MONTH    =        3.00
                                   BAL-3+MONTH     =        4.00
                                   TURNOVER        =     5555.00
                                   CLAY-VALUE      =        3.00
                                   INDUSTRY        = san
                                   DAYS-ALLW-ON-N/M= 1
                                   AGENT-CODE      =
                                   TYPE            =
                                   CURRENCY        = $
LOG RECORD NO   =          70          ALPHA-CODE      = NORD
DATE            =    31/08/88      New CREDIT-LIMIT    =      966.0
TIME            = 10:22:35.2       Old CREDIT-LIMIT    =        0.0
LOG NUMBER      =          13
LDEV NUMBER     =          71
DST/TRANS NO    =  1037/   3
SET RECORD NO   =         907
IMAGE CALL      = DBUPDATE
DATASET NAME (M)= CUSTOMER
PROGRAM NAME    = QUERY.PUB.SYS
USER NAME       = TEST.PROGRAM,DAVE

LOG RECORD NO   =          72      DUMMY1          = XXXX
DATE            =    31/08/88      NEXT-BATCH-NO   =       13.00
TIME            = 10:24:09.2       NEXT-INV-NO     =        1.00
LOG NUMBER      =          13      VAT-RATE        =       15.00
LDEV NUMBER     =          71
DST/TRANS NO    =  1037/   5
SET RECORD NO   =           1
IMAGE CALL      = DBDELETE
DATASET NAME (M)= REFERENCE
PROGRAM NAME    = QUERY.PUB.SYS
USER NAME       = TEST.PROGRAM,DAVE
```

```
<><> ANALYZE ₵ Logfile Enquiry        (Run: SUN, SEP 17, 1989,  4:29 PM)
From 08:28 30/08/88 until 17:39 31/08/88   Database SALESM.DAVE.PROGRAMS
User @.@                                    Process QUERY.@.@
LDev @                                      Logfile MSLOG.DAVE.PROGRAMS
Set  @                                      Item   @=@
-----------------------------------------------------------------------

LOG RECORD NO  =          74      DATABASE NAME  = SALESM.DAVE.PROGR
DATE           =    31/08/88      ABORT          = NO
TIME           = 10:24:21.8       TIME OPENED    = 10:20 31/08/88
LOG NUMBER     =          13
LDEV NUMBER    =          71
DST/TRANS NO   =  1037/
SET RECORD NO  =
IMAGE CALL     = DBCLOSE
DATASET NAME   =
PROGRAM NAME   = QUERY.PUB.SYS
USER NAME      = TEST.PROGRAM,DAVE

<><><><><><><><>     End of Log Enquiry Option     <><><><><><><><><><>
```

The heading of the enquiry is similar to that of the audit trail option. In addition we have the following criterion:

Item

> The data item condition selection criterion. In the above example this was @=@ indicating that all data item values will be selected (the default ₵ i.e. all items equal to all values). Other examples could be AMOUNT < 100, ACCOUNT = JONES etc.

Detailed definitions of each part of the body of the enquiry follow:

LOG RECORD NO =

> Each entry in a log file is allocated a sequential number, which is written into the log entry itself. This number also represents the actual record number of the log entry in the file. Log files have a fixed record width of 128 words, however one DBPUT or DBDELETE may well be for an entry with an entry size of more than 128 words. In this case continuation records are used and thus you may see the Log Record No jump from 136 to 140 due to 4 log records being used for a single DB call. In addition, log records are written for purposes other than DB entries (a log process starting or stopping for example). These records are not displayed by ANALYZE as they are not normally of interest. Their details, however, are read and are used to match log entries to users and processes.

DATE   =

> The date on which this DB Call was made. This date can be formatted in 3 different ways (see above).

TIME   =

> The time at which this DB Call was made. The format of this time is hh:mm:ss.t, where hh is hour (in 24 Hour format), mm is minutes, ss seconds and t tenths of seconds. Unfortunately, the transaction logging process only records times to the nearest tenth of a second, which is longer than many DB Calls take to complete. Thus this time cannot be fully used to directly measure performance.

LOG NUMBER   =

> When the logging process is started each process which initiates logging (by calling DBOPEN) is allocated a sequential log number starting from 1. Thus all entries with the same log number within the same logfile have been made by the same process (user/program combination).

LDEV NUMBER   =
> The logical device upon which this database call was made. Each process (as opposed to a program) can run upon only one logical device number. Batch jobs usually will have logical devices of 10 or less (depending on your particular configuration of I/O devices).

DST/TRANS NO   =
> In addition to the log number each separate DBOPEN call within a process is allocated a Data Segment Number. It is this number which is placed in the first two bytes of the Database Name in the DBOPEN call. Each DST NO/LOG NO combination will be unique within a log file (unless the file has been corrupted) and represents a specific DBOPEN within a specific process. More than one process may have the same DST NO, but they will each have separate LOG NOs.
>
> Within each DBOPEN itself there are a series of database transactions. Unless bracketed by a DBBEGIN/DBEND pair, each individual database call is a different transaction and will be allocated a sequential number starting from one. Logically connected calls may, however, be surrounded by calls to DBBEGIN and DBEND. These calls then all become one single transaction and are all given the same transaction number.

SET RECORD NO =
> Each DB Call relates to a single entry within the relevant dataset. This entry is identified in the log file by its record number within the set at the time the call was made. This is the same record number as is passed back in the Status array of each Db Call. ANALYZE uses this number to attempt to retrieve the key values of detail set DBUPDATES (see above).

IMAGE CALL   =
> The following types of call are logged by IMAGE or TurboIMAGE:

>> DBDELETE
>> DBPUT
>> DBUPDATE
>> DBBEGIN
>> DBEND
>> DBMEMO
>> DBOPEN
>> DBCLOSE

> Different details are displayed in the enquiry for each of these calls. DBDELETE and DBPUT will be accompanied by the full item list for this set, with each item's value (including sub-items). DBUPDATE will have the Primary Key Value and each item updated (with both old and new values). DBBEGIN, DBEND and DBMEMO will have the Text associated with them. DBOPEN and DBCLOSE have their own information (see below).

DATASET NAME (D)=
> Each DB Call relates to a single entry within a specific dataset. This may be a Detail (D), or a Manual Master (M) or an Automatic (A). However, as their name implies, Automatic Masters are not explicitly amended or deleted by users. Instead, IMAGE automatically maintains their values and so changes to Automatic Masters are not logged (only the changes made to their details are logged).

PROGRAM NAME   =
> The name of the program running in the process (user/program combination) which performed the DB Call. This name will be fully qualified.

USER NAME   =
> The name of the user running the process (user/program combination) which performed the DB Call. This name will be fully qualified with the Account Name, followed by the Group Name into which the user was logged.

ALPHA-CODE   =
> Calls to DBPUT, DBUPDATE and DBDELETE will be accompanied by a list of data items within the dataset (in this example it is the data item Alpha-Code), followed by their data values. The full data item value will be displayed (unless truncated by the page width). Sub-items will also be displayed separately; thus AMOUNT(1) is the name of the first sub-item of the data item AMOUNT. DBPUT and DBDELETE will display every data item in the set. DBUPDATE will display only the Primary Key Value and the old and new values of the data items that are changed.

DATABASE NAME   =
> The name of the database DBOPENed or DBCLOSEed. This is displayed for confirmation only as ANALYZE will only perform enquiries for one database at a time. It is perfectly possible for more than one database to be logging to the same logfile, in which case the other databases are simply ignored.

SECURITY LEVEL   =
> The access level associated to the password supplied by the program opening the database. A level of 64 indicates Creator access.

USER ID   =
> A call to DBOPEN can specify a User Id so as to differentiate one user with the same logon name from another. This is done by adding "/user id" to the Database name in the DBOPEN call (see TurboIMAGE Manual). In QUERY, for example, one may enter SALES/EXPORT.DBS as the database name. The User Id EXPORT would then be logged in the log file DBOPEN record. The only place that this Id is used is in the transaction logfile itself.

DBOPEN MODE   =
> The mode in which DBOPEN was called which is a number from 1 to 4 (Modes 5 to 8 are read only and are not logged).

New CREDIT-LIMIT   =
> Calls to DBUPDATE log the old and new values of each data item whose values actually change. The enquiry option will detail each item or sub-item which has changed and display its newly updated value.

Old CREDIT-LIMIT   =
> Calls to DBUPDATE log the old and new values of each data item whose values actually change. The enquiry option will detail each item or sub-item which has changed and display its old value prior to the update call.

ABORT   =
> A DBCLOSE may be explicitly called in a program, or may be implicitly executed by MPE when a program terminates (either aborted in the middle or simply ended without calling DBCLOSE). In the latter case an Abort marker is logged in the DBCLOSE log entry indicating that a potential abort has occurred. However, there is no way of distinguishing between a deliberate abort (i.e. the user broke the program, or the datacomm line dropped out, or the program crashed) from a program simply finishing without calling DBCLOSE explicitly. The implicit DBCLOSE issued whenever a program terminates with an open database is considered as an abort by the logging process.

TIME OPENED   =
> The time and date of the DBOPEN is displayed when each DBCLOSE entry is displayed. Thus, one can see how long the database had been opened.

> Note: numeric data items with a length of more than 10 words (such as those used by Omnidex, e.g. J253) cannot be correctly displayed. In these cases their values are replaced by the message "Unknown Data Type ₡ Data Not Displayed." In addition, nonprintable characters are replaced by the "." character.

5.6 LOG STATISTICS OPTION

The purpose of this option is to produce a wide range of statistical figures about a logfile (or a series of continuation logfiles). In addition, these figures may be accumulated into a History File for the database so that an historical picture of the actual use of the database can be produced.

If the program is run by the Creator of the database in the same group and account as the database the following question will be asked:

```
OK to Update Database History file (Y/N) >>
```

The statistics for this database can be accumulated into a history file which is only creatable or modifiable if the user is the creator of the database. This file will be used by the DIAGNOSE module enabling, in the future, more accurate recommendations to be made based upon actual usage of the database. Enter Y to store this run's set of statistics in the History File. Enter N if you do not wish to do so.

The recommended procedure is to perform this function as part of the log cycling procedures. Typically a database will be fully backed up (using STORE or DBSTORE) at the same time as the old log file(s) are purged and a new log file created. This is the perfect time to analyze the old log file(s) and store their details in the History File. In this way the full details of each file will be available. It is possible to run ANALYZE at any time, even on a currently open logfile but if the History File is updated half-way through the log cycle (see Appendix G) the later transactions' statistics will not be stored. This is because ANALYZE prevents the same log file being updated more than once into the History File.

Data on this logfile will also be stored in the FXSTAT.PUB.HQ trends database if an analyze statistics run is perfomed, but only if the user selects OK to update History File.

If the user is not the creator of the database the above question does not appear and the statistics processing starts immediately. There are no selection criteria in the Statistics option. The entire log file, or set of logfiles, is processed so that a full picture of this database's modifications can be established. The following example lines will appear on the screen (indicating the progress of the selection process):

```
Reading Log File....................
 22:28 25/08/88 <><>  Logid DSTSTLOG Started    <><>
 18:29 21/08/88 <><>  Logid DSTSTLOG Restarted  <><>
 13:42 22/08/88 <><>  Logid DSTSTLOG Restarted  <><>
      296 entries selected
```

The number of entries selected will be only those records which relate to the selected database. This may not be the same number as the total number of log records in the log file. It may be less (in the case of multiple databases logging to the same file), or more (in the case of continuation logfiles).

The program then outputs the statistical reports for this logfile. On the screen the following message appears:

```
Producing User & Set Statistics.....
```

Upon finishing the current log file's report the program will attempt to open the database history file. On the screen the following message appears:

```
Reading database history file.......
```

If this file exists the latest figures stored in it will be output. If the file does not exist a message will appear on the screen to that effect.

5.7 EXPLANATION OF STATISTICS OUTPUT

An example of the output from the Statistics option follows. The formal file designator for this report is FXSTAT, and this output may be re-directed by use of a file equation. The example below is for a record width of 132 characters, but a maximum width of 256 characters is possible.

Part one of the statistics produced relates to users and programs:

```
<><> ANALYZE ₵ Logfile Statistics      (Run:  FRI, SEP 29, 1989, 11:15 PM)    Page:  1
From 22:14 28/09/89 until 22:11 29/09/89   Database MMOVE.WORKS.PLC        Logfile MMLOG.WORKS.PLC
UserProcess Statistics for this Logfile:
```

User Name Num and Group Abends	Program Name	Time Opened	Time Closed	Num Puts	Num Deletes	Num Updates	Num Null Updates	Num Opens	Num Closes	Num Aborts	Num Begins	Num Ends
29/09/89												
PMPROD.PLC,WORKS 0	MMMENUP.WORKS.PLC	08:09	08:15	0	0	0	0	1	1	1	0	0
SALESB.PLC,SALES 0	BPMEN2P.SALES.PLC	06:51	08:36	10	0	15	0	1	1	1	0	0
SALESB.PLC,SALES 0	MMMENUP.WORKS.PLC	08:36	08:38	2	0	4	0	1	1	1	0	0
PR.PLC,SALES 0	BPMENUP.SALES.PLC	09:06	09:07	2	0	2	0	1	1	0	0	0
PR.PLC,SALES 0	BPMENUP.SALES.PLC	09:07	09:23	12	0	13	0	1	1	0	0	0
PROGS.PROGRAM,TESTSM 0	INCLUDPP.TESTSM.PROGRA	09:48	09:50	0	0	0	0	1	1	1	0	0
PR.PLC,SALES 0	BPMENUP.SALES.PLC	09:23	09:52	23	0	34	0	1	1	0	0	0
NDLAB.TECH,QC 0	MMMENUP.WORKS.PLC	10:47	10:57	0	0	0	0	1	1	0	0	0
SALESB.PLC,SALES 0	BPMEN2P.SALES.PLC	08:38	12:24	17	0	48	0	1	1	1	0	0
SHAUGH.PLC,SALES 0	BPMEN2P.SALES.PLC	08:02	14:05	43	0	122	0	1	1	1	0	0
SALESB.PLC,SALES 0	BPMEN2P.SALES.PLC	14:05	14:15	2	0	3	0	1	1	1	0	0
SALESB.PLC,SALES 0	BPMEN2P.SALES.PLC	12:24	14:17	6	0	34	0	1	1	1	0	0
SALESB.PLC,SALES 0	MMMENUP.WORKS.PLC	14:17	14:27	5	0	10	0	1	1	1	0	0
SALESB.PLC,SALES 0	BPMEN2P.SALES.PLC	14:27	15:00	2	0	13	0	1	1	1	0	0
SALESB.PLC,SALES 0	MMMENUP.WORKS.PLC	15:00	15:01	2	0	4	0	1	1	1	0	0
SALESB.PLC,SALES 0	BPMEN2P.SALES.PLC	15:01	15:06	2	0	8	0	1	1	1	0	0
PR.PLC,SALES 0	BPMENUP.SALES.PLC	14:51	15:15	20	0	38	0	1	1	0	0	0
SALESB.PLC,SALES 0	MMMENUP.WORKS.PLC	15:07	15:32	1	0	2	0	1	1	1	0	0
SALESB.PLC,SALES 0	BPMEN2P.SALES.PLC	15:32	15:34	0	0	5	0	1	1	1	0	0
SALESB.PLC,SALES 0	MMMENUP.WORKS.PLC	15:34	15:35	1	0	2	0	1	1	1	0	0
SHAUGH.PLC,SALES 0	BPMEN2P.SALES.PLC	14:16	15:42	4	0	16	0	1	1	1	0	0
SALESB.PLC,SALES 0	BPMEN2P.SALES.PLC	15:35	15:49	0	0	5	0	1	1	1	0	0
SALESB.PLC,SALES 0	MMMENUP.WORKS.PLC	15:49	15:52	2	0	4	0	1	1	1	0	0
PMSTORES.PLC,WORKS 0	STMENUP.WORKS.PLC	15:45	15:56	0	0	0	0	1	1	1	0	0
SALESB.PLC,SALES 0	BPMEN2P.SALES.PLC	15:52	15:59	0	0	0	0	1	1	0	0	0
NDLAB.TECH,QC 0	MMMENUP.WORKS.PLC	16:25	16:26	0	0	0	0	1	1	Aborts	Begins	Ends
NDLAB.TECH,QC 0	MMMENUP.WORKS.PLC	16:27	16:29	0	0	0	0	1	1	0	0	0
NDPROD.PLC,WORKS 0	MMMENUP.WORKS.PLC	06:33	17:08	78	0	172	0	1	1	0	0	0
CWPROD.PLC,WORKS 0	MMMENUP.WORKS.PLC	10:06	17:14	12	0	9	0	1	1	0	0	0
PMPROD.PLC,WORKS 0	MMMENUP.WORKS.PLC	08:16	21:31	137	1	252	0	1	1	0	0	0
PMPROD.PLC,WORKS 0	MMMENUP.WORKS.PLC	08:04	22:11	69	0	139	0	1	1	1	0	0
TOTALS 0				452	1	954	0	31	31	20	0	0

The first statistical report is the full list of user processes which have opened the specified database with write access. Definitions of each figure reported follow:

User Name and Group

> The fully qualified name of the user running the program which opened the database with write access. The format is User.Account,Group.

Program Name

> The fully qualified name of the program which opened the database. The format is Name.Group.Account. If this name is too long it will be truncated.

Time Opened

> The time, in 24 hour hh:mm format, at which the call to DBOPEN was executed. The date will have been output previously in the User Name column (and relates specifically to the time closed).

Time Closed

> The time, in 24 hour hh:mm format, at which this database was closed. The date will have been output previously in the User Name column. This report is, in fact, sorted by the date and time of the DBCLOSE (DBRECOV will produce a report sorted by the time of the DBOPEN.) A value of 00:00 indicates that this process has not closed the database yet. This could be due to the logging process still running, or a full logfile with change logging not enabled. It could also be the result of a system failure causing all active processes to terminate without closing the database.

Num Puts

> The total number of DBPUTs issued by this DBOPEN/User/Program to any set within the specified database.

Num Deletes

> The total number of DBDELETEs issued by this DBOPEN/User/Program to any set within the specified database.

Num Updates

> The total number of all DBUPDATES issued by this DBOPEN/User/Program to any set within the specified database (including those updates to automatic master sets associated with puts or deletes of detail set entries).

Num Null Updates

> The total number of DBUPDATES issued by this DBOPEN/User/Program to any set within the specified database which did not actually change any data item values. This statistic is very important to program performance. It represents unnecessary database calls, and thus redundant I/O. This quantity is also included in the Num Updates figure above, and thus comparison of these two figures gives an indication of the potential for performance improvement.

Num Opens

> The total number of DBOpens issued by this DBOPEN/User/Program to any set within the specified database. This figure should always be one, unless the logfile is corrupted.

Num Closes

> The total number of DBCLOSEs issued by this DBOPEN/User/Program to any set within the specified database. Again, this figure should always be one unless, as above, the user/program has not yet closed the database.

Num Aborts

> The total number of program aborts detected by the logging process. A program abort is defined as a process which ended without explicitly calling DBCLOSE. Ideally this figure should be zero, but could be one if the program terminated abnormally. However, it could also be one if

the program simply failed to call DBCLOSE before ending. There is no known way of differentiating between these two events.

Num Begins

The total number of DBBEGINs issued by this DBOPEN/User/Program to the specified database. A DBBEGIN signifies the start of a logical transaction (see TurboIMAGE manual and Appendix G). ANALYZE recongnizes the log file records for DBXBEGIN, but as these relate to multiple database transactions, they are not added to this total.

Num Ends

The total number of DBENDs issued by this DBOPEN/User/Program to the specified database. A DBEND signifies the end of a logical transaction (see TurboIMAGE manual and Appendix G). The number of DBENDs should equal the number of DBBEGINs. If it is less then there are some incomplete logical transactions in the logfile. This can be the case if this option is run during an active logging process (i.e., some users may still be in mid-transaction), or if a system failure or program abort has occurred. In the latter case these incomplete transactions will not be recovered when using DBRECOV roll-forward recovery. ANALYZE recognizes the log file records for DBXEND, but as these relate to multiple database transactions, they are not added to this total.

Num Abends

The total number of Abnormal transaction ends caused by this DBOPEN/User/Program in the specified database. In this case the program has issued a DBBEGIN to start a logical transaction but has failed to issue a DBEND before finishing. Ideally this figure should be zero, but if the program aborted the log process would automatically post Abend entries signifying an abnormal end. Such incomplete transactions will not be recovered with DBRECOV.

Each of the above figures is quoted for the individual process and also totalled for the entire logfile(s).

Part two of the statistics relates to datasets within the database:

```
<><> ANALYZE C Logfile Statistics     (Run: FRI, SEP 29, 1989, 11:15 PM)
From 22:14 28/09/89 until 22:11 29/09/89   Database MMOVE.WORKS.PLC
Logfile MMLOG.WORKS.PLC                    Page:    1
---------------------------------------------------------------------


Dataset Statistics for this Logfile:

                       Num       Num       Num      Num Null
Dataset Name           Puts    Deletes   Updates   Updates

DESTINATIONS             0         0         0         0
SOURCES                  0         0         0         0
JOURNEYS                 0         0         0         0
MATERIALS                0         0         0         0
MOVEMENTS              442         1       157         0
STOCKS                   2         0       319         0
MATERIAL-MASTER          0         0         0         0
MATERIAL-PLANSET         8         0       462         0
COSTING-PERIOD           0         0         0         0
MATERIAL-FORMULA         0         0         0         0
SYSTEM-DESC              0         0         0         0
SALES-PLANSET            0         0         0         0
ROYALTY-STOCKS           0         0        16         0
MATERIAL-LOSSES          0         0         0         0


TOTALS                 452         1       954         0
```

The second statistical report lists each dataset in the specified database. Definitions of each figure reported are as follows:

Dataset Name

>The name of the dataset. This will be a full list naming each set in the database, unless the password used to open the database did not give full read access to each set in the database. In this case the name will read "Unknown set."

Num Puts

>The total number of DBPUTs issued to this dataset by all users and programs in this logfile(s).

Num Deletes

>The total number of DBDELETEs issued to this dataset by all users and programs in this logfile(s).

Num Updates

>The total number of DBUPDATES issued to this dataset by all users and programs in this logfile(s). For automatic masters this figure represents the number of puts and deletions to associated detail sets which would, at minimum, cause an update of the automatic master's chain head pointers. Such puts and deletions could also cause puts and deletions to the automatic itself. It is not possible to detect these occurrences from the logfile itself, and thus all detail puts & deletions are considered as causing updates to the automatic master. This gives the advantage of identifying write access to automatic masters as well as manual masters.

Num Null Updates

>The total number of DBUPDATES issued to this set which did not actually change any data item values. This statistic is very important to database performance. It represents unnecessary database calls, and thus redundant I/O. This quantity is also included in the Num Updates figure above, and thus comparison of these two figures gives an indication of the potential for performance improvement. Comparison of this figure with the same figure for users and programs should isolate the sections of these programs which are being inefficient.

>Each of the above figures is quoted for the individual sets and also totalled for the entire logfile(s). These totals should agree with the totals for user processes.

>Part three of the statistics is a report of the contents of the database History file (if any). If no file exists this report will not be produced. If the user is the creator and has answered Y to the question "OK to Update History File" above then the History file will be created and then reported. In this latter case the figures should agree with the previous two sections. If the History file existed previously then the figures produced refer to the historical log files which have been accumulated in the History file (and thus the figures will be higher than those for the current logfile).

Database MMOVE.WORKS.PLC   History File MMOVEFX.WORKS.PLC
Overall Historical Statistics:
Start Date/Time of History File = 22:14 28/09/89
End   Date/Time of History File = 22:11 29/09/89

Maximum Concurr Users	Num Puts	Num Deletes	Num Updates	Num Gets	Num Opens	Num Closes	Num Aborts	Num Abends	Num Days	Maximum Log Recs
7	452	1	954	955	31	31	20	0	1.0	2002

Max Puts Per Day	Max Deletes Per Day	Max Updates Per Day	Max Gets Per Day
452.0	1.0	954.0	955.0

Hourly Statistics:

Hour	Num Puts	Num Deletes	Num Updates	Num Opens	Num Closes	Num Aborts	Num Abends
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0	0	2	0	0	0
7	0	0	0	0	0	0	0
8	34	0	54	6	3	3	0
9	93	0	163	4	4	1	0
10	48	0	102	2	1	0	0
11	19	0	54	0	0	0	0
12	6	0	19	1	1	1	0
13	65	0	158	0	0	0	0
14	61	0	154	5	4	4	0
15	37	0	84	9	12	10	0
16	0	0	0	2	2	0	0
17	0	0	0	0	2	0	0
18	0	0	0	0	0	0	0
19	17	0	34	0	0	0	0
20	33	0	66	0	0	0	0
21	39	0	66	0	1	0	0
22	0	0	0	0	1	1	0
23	0	0	0	0	0	0	0

Overall Historical Statistics:

Dataset Statistics:

	Num Puts	Num Deletes	Num Updates	Num Gets	Num Null Updates	Max Increase
DESTINATIONS	0	0	0	0	0	0
SOURCES	0	0	0	0	0	0
JOURNEYS	0	0	0	0	0	0
MATERIALS	0	0	0	0	0	0
MOVEMENTS	442	1	157	158	0	441
STOCKS	2	0	319	319	0	2
MATERIAL-MASTER	0	0	0	0	0	0
MATERIAL-PLANSET	8	0	462	462	0	8
COSTING-PERIOD	0	0	0	0	0	0
MATERIAL-FORMULA	0	0	0	0	0	0
SYSTEM-DESC	0	0	0	0	0	0
SALES-PLANSET	0	0	0	0	0	0
ROYALTY-STOCKS	0	0	16	16	0	0
MATERIAL-LOSSES	0	0	0	0	0	0
TOTALS	452	1	954	955	0	451

Program Statistics:

	Num Puts	Num Deletes	Num Updates	Num Gets	Num Null Updates
MMMENUP.WORKS.PLC	309	1	598	599	0
BPMEN2P.SALES.PLC	86	0	269	269	0
BPMENUP.SALES.PLC	57	0	87	87	0
INCLUDPP.TESTSM PROGRAM	0	0	0	0	0
STMENUP.WORKS.PLC	0	0	0	0	0
TOTALS	452	1	954	955	0

The third statistical report is a summary of statistics from the History File for the specified database. Definitions of each figure reported are as follows:

Start Date/Time of History File
> The date and time of the beginning record in the first logfile entered into the History file.

End Date/Time of History File
> The date and time of the last record in the last logfile entered into the History file. These dates are present to indicate the breadth of time which this History file covers.

Maximum Concurr Users
> This figure is the largest observed number of concurrently operating write users at any one time. This occurrence may not have been in the current logfile, but represents an historical maximum. This figure only represents write users, as transaction logging does not record read-only users (DBOPEN modes 5-8). Thus, this figure is not a complete figure for the maximum number of users on the database concurrently. However, by multiplying this figure by a factor of, say, 1.5 an approximate value can be reached. This multiplying factor will, of course, be different for each database. It should be noted that programs which only read the database can still open the database in modes 1 to 4 and thus be logged. In the example figures above the program STMENUP.WORKS.PLC did not issue any DBPUTs, DBDELETEs or DBUPDATES but still opened the database in mode 1 and was thus logged as a write user.

Num Puts
> The total number of DBPUTs issued to this database by all users and programs in all the logfiles whose figures have been added to this History file.

Num Deletes
> The total number of DBDELETEs issued to this database by all users and programs in all the logfiles whose figures have been added to this History file.

Num Updates
> The total number of DBUPDATES issued to this database by all users and programs in all the logfiles whose figures have been added to this History file.

Num Gets
> Although neither read access to the database nor DBGET calls are logged a certain amount of DBGETs can be implied from the preceeding figures. For a DBUPDATE or DBDELETE to have been issued successfully, they must have been preceded by at least one DBGET. Of course, there may have been many more (chained reads of detail sets, or serial reads) but we can be certain that at least one was made. Thus, this figure is simply the sum of the minimum number of DBGETs preceding DBUPDATES and DBDELETES. The minimum number of DB calls made is thus the sum Num Puts + Num Deletes + Num Updates + Num Gets. Although more calls undoubtedly have been made this figure is the minimum there must have been.

Num Opens
> The total number of DBOPEN calls issued to this database by all users and programs in all the logfiles whose figures have been added to this History file.

Num Closes
> The total number of DBCLOSEs issued (explicitly by user programs, or implicitly by MPE) to this database by all users and programs in all the logfiles whose figures have been added to this History file.

Num Aborts
> The total number of program aborts without calling DBCLOSE explicitly (see definition of aborts above) caused on this database by all users and programs in all the logfiles whose figures have been added to this History file.

Num Abends

> The total number of abnormal terminations of logical transactions (see definition of Abend above) for this database by all users and programs in all the logfiles whose figures have been added to this History file.

Num Days

> The total number of days of transaction logfiles whose statistics have been added to this History file. To get an average number of Puts per day for this database, for example, one would divide the Num Puts figure by the Num Days figure. This average can then be compared with the maximum figure quoted later.

Maximum Log Recs

> A figure representing the largest number of log entries relating to this database selected in any one run of the Statistics Option of ANALYZE. This will not be the total of Puts+Deletes+… etc., because each call may span many log records. This figure gives a feel for the potential size of logfiles created.

Max Puts per day

> For each logfile the total number of puts is divided by the number of days for which the logfile ran. The maximum observed value of this figure is stored in the History file. When compared with the average number of puts (calculated by dividing the total days into the total puts) the volatility of the put access to the database can be judged. A large maximum with a small average indicates an abnormally large addition run which is run infrequently. Comparable numbers indicate a smoother process of adding data entries.

Max Deletes per day

> For each logfile the total number of deletes is divided by the number of days for which the logfile ran. The maximum observed value of this figure is stored in the History file. When compared with the average number of deletes (calculated by dividing the total days into the total deletes) the volatility of the delete access to the database can be judged. A large maximum with a small average indicates an abnormally large deletion run which is run infrequently. Comparable numbers indicate a smoother process of deleting data entries.

Max Updates per day

> For each logfile the total number of updates is divided by the number of days for which the logfile ran. The maximum observed value of this figure is stored in the History file. When compared with the average number of updates (calculated by dividing the total days into the total updates) the volatility of the update access to the database can be judged. A large maximum with a small average indicates an abnormally large update run which is run infrequently. Comparable numbers indicate a smoother process of updating data entries.

Max Gets per day

> For each logfile the total number of gets (as defined above) is divided by the number of days for which the logfile ran. The maximum observed value of this figure is stored in the History file. When compared with the average number of gets (calculated by dividing the total days into the total gets) the volatility of the write access to the database can be judged.

Hourly Statistics

> The same definitions for Num Puts, Num Deletes etc. are used as for the total numbers for every logfile stored in the History file (the first line of statistics in this section). However, these figures are now analyzed by the hour of the day in which they were made. Twentyfour lines of statistics follow, one for each hour of the day starting at zero for Midnight to 12:59 etc. Thus, one can see which hour of the day gets the most, or least, write transactions or aborts etc.

Dataset Statistics

> Again, the same definitions for Num Puts, Num Deletes, Num Updates and Num Gets are used as for the total numbers for every logfile stored in the History file (the first line of statistics in this section). This time the total figures are analyzed by the dataset to which they refer. Thus the most

heavily used datasets can be identified, this may help in planning dataset logical device placement.

In addition the Null Updates total is also noted to identify inefficient I/O access. The Max Increase figure is the maximum observed increase in the number of entries in each set for a logfile. This indicates the minimum number of free entries each set should have to avoid filling up. A set's Capacity should be at least this number of entries greater than the current entry count (and indeed probably much more C DIAGNOSE will use this, and other figures, in future releases).

Program Statistics

Again, the same definitions for Num Puts, Num Deletes, Num Updates and Num Gets are used as for the total numbers for every logfile stored in the History file (the first line of statistics in this section). This time the total figures are analyzed by the programs which made the calls. Thus the heaviest write access program can be identified. Identifying these heavy programs will help in performance optimization. Optimizing only these programs should benefit performance.

5.8 ANALYZE ERROR OR WARNING MESSAGES

History File SALESMFX.DAVE.PROGRAM does not exist, or is not available

While running the Logfile Statistics option, ANALYZE has tried to open the database History file. This file has the same name of the database suffixed by "FX," and has a file code of 6598. If this file does not exist, or sufficient access to this file is not possible then this message will appear. The program will continue, but will not print out the summary of the History file.

If an unexpected program error occurs, such as a bounds error etc., the program will abort with the following type of message:

```
<><><><><>  Program Error Occurred in Module TIMEINPUTOK  <><><><><>
 STRING INDEX EXCEEDS CURRENT LENGTH (PASCERR 651)
 LogFile Rec =      294
 Program Module Trace:
      OUTERBLOCK
      AUDIT_TRAIL
      AUDIT_CRITERIA
      AUDITCRITERION
      TIMEINPUTOK
 LogNo       =          0 User        =
 DSTNo       =          0 ProgName    =
 Ldev        =          0 DBName      =
 Logid       =            LogType     = Trailer
 DBType      = Unknown  LogFileName = XYLOG.PUB.HQ
 Stack Marker:
      Index Register  %000000 (Decimal      0)
      Code Location   %004550 (Decimal   2408)
      Status          %060004 (Decimal  24580)
      Delta-Q         %000027 (Decimal     23)

To report this fault please send the following to Open Seas:
 1) Current Schema File of Database,
 2) Copy of this Logfile, and
 3) This printout
<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
```

Such an abort has not been expected, and as such represents a problem either with the software, the logfile or database itself. It will require further investigation, for which purpose the three items listed above should be sent to Open Seas.

"filename" is not a valid IMAGE/TurboIMAGE Transaction Logfile.

Any file entered as a database log file must both exist and have a file code of 1090. If either of these conditions are not met then this message is displayed and the user must re-enter the name. If a valid logfile has been entered but it is empty (i.e. has no records in it at all) then this message is also displayed.

[Invalid option] or [INVALID RESPONSE C PLEASE RE-ENTER]

> In FXUTIL and ANALYZE there are some questions where there is an allowable range of acceptable answers (e.g., U, L or P for User, Ldev or Process). If an invalid option has been entered this message is displayed and the user must re-enter one of the allowed values.

Error in DBGET on Database "dname", Set Number "n"

> ANALYZE has to access detail sets where DBUPDATES have been performed so as to establish the Primary Key values. Any error encountered in reading the detail set will result in this message (should never occur unless there is a problem with the database). Contact Open Seas if this error occurs.

Error Opening Database

> For example the database does not exist! The DBERROR message will also be displayed to indicate the reason. Usually one has simply mis-typed the name or password, or perhaps one does not have MPE security access to the database.

Error Closing Database

> This error should never occur. If it does contact Open Seas.

Error C too many consecutively open databases
Program will continue, some data will be lost

> There is a maximum number of concurrently open databases in a log file that ANALYZE will handle. If this value is exceeded the following data will not be fully analyzed and data will be lost. The program will, however, attempt to continue processing so as to report what it can. For MPE/V systems this limit is 63. For MPE/iX systems the limit is 4096.

Error C too many concurrent user processes
Program will continue, some data will be lost

> There is a maximum number of concurrently running user processes (with write access) that ANALYZE will handle. If this value is exceeded the following data will not be fully analyzed and data will be lost. The program will, however, attempt to continue processing so as to report what it can. For MPE/V systems this limit is 63. For MPE/iX systems the limit is 1024.

Exclusive access denied to file
"dnameFX.group.account"; Statistics for this
logfile not stored, run again as CREATOR of "dname"

> In the statistics option a user has attempted to update or create the History file. Only the creator is allowed to do this. Again, this message should never occur but if it does contact Open Seas with details of the user and creator and group.

Incorrect Password C please re-enter.

> A database password must be entered to open the database at the start of ANALYZE. If this password is incorrect then this message is displayed. Simply re-enter the password.

> Also, some logging processes have passwords allocated to them and this password is stored in the root file of the database (via DBUTIL). In this case ANALYZE will ask for the password and the user must enter this before gaining access to the log files. If the password is incorrect this message will be displayed.

Unrecoverable error encountered in Job Mode, program terminated.

> An error has occurred in a job file in entering a selection criterion or menu choice. If the program was running as an interactive process the user will simply be re-prompted for a valid entry. However, if this occurs in job mode the program has no option but to terminate.

Continuation logfile not found in Job Mode, selection terminated.

> Continuation logfiles are automatically detected and opened by ANALYZE. However, if one of the files in a set cannot be opened (such as it being missing for example) then in an interactive session the user will be prompted for the name of the continuation logfile. In job mode, however, the program has no-one to ask and so terminates.

Database name in history file does not match, no update done.

The history file created by ANALYZE contains the fully qualified name of the database. This is checked against the name of the currently opened database before an update can be done. The above message indicates a mis-match. Check that this file is a valid ANALYZE history file C if not, purge the file and re-run the statistics as the creator.

5.9 STARTING ANALYSIS IN MIDDLE

It is not normally possible to run ANALYZE on only part of the log file sequence, i.e. all log files in a set must be present and the name of the log file supplied to ANALYZE must be the first one in the sequence. This is necessary because information concerning part of a transaction may reside in a previous log file. However if you set JCW FXANAL01 to one, you can start the analysis on a later log file. In this case any transactions commenced in the current or later log files will be included in the analysis, otherwise the transactions will be ignored.

SECTION 6 ₵ CAPACITY FUNCTION

The purpose of this module is to allow capacity changes to be made to sets with maximum convenience and speed.

```
Requirements & Pre-conditions:
•    Ensure that a back-up copy of the database is available.
•    The user must be logged on as the database creator.
•    Exclusive access to the database (mode 3) is required.
•    MPE read, write, and save access to both the log-on group and the group in which
     the database resides is required.
```

For a simple capacity change the following is the dialog between the user and the FLEXIBASE software:

```
:RUN FLEXIBAS.PUB.HQ
```

Select option 5 from the menu.

```
Database name >>
```

Enter the name of the database in which the set capacity change is required. The user must be the creator of the database and the software requires exclusive access to the database (except when setting up a job file).

```
Set name >>
```

Enter the name of the set for which the capacity is to be changed.

```
Number of entries is nnnnn
Old capacity is nnnnn  (nn% full)
```

The program reports the existing capacity, the number of entries present in the dataset and the percentage full.

```
Prime numbers are no better than others but ODD
numbers are safer than EVEN
```

For master sets, the user is reminded that prime numbers are not considered useful for selecting suitable capacities (irrespective of recommendations from other sources to that effect), but that ODD numbers will avoid possible inefficiencies associated with capacities equal to powers of 2.

```
Enter new capacity (or change %) (or valid formula to give capacity)>>
```

Enter the new desired capacity. This may be done either by entering a specific number (e.g. 12300), as a percentage change up or down (e.g. 10%, +10%, or -10%), or as a formula (e.g. HWM * 2). See the Section entitled "Automatic Capacity Management" for the possible types of formula and the syntax required.

```
Capacity (detail sets are blocked-up) will be changed to nnnn (83% full)
or
Capacity will be changed to nnnnn (83% full)
```

Where a percentage change, or a formula, has been specified the user is informed of the new chosen capacity. Detail set capacities are always rounded-up to a multiple of the blocking factor (i.e. blocked-up).

```
Do you want to proceed ₵  Enter Y or N >>
```

The software asks for confirmation before physically making the change.

```
Set being processed.....
```

The required changes are currently being actioned.

```
Percentage of secondaries = 23%
<><><> CAPACITY CHANGE COMPLETE <><><>
```

The required changes are confirmed to be complete.

The percentage of entries which are synonyms (or secondaries), as opposed to being placed in their primary position, is quoted for information. Synonym percentages up to 30% are fairly normal. Percentages above 40% might begin to raise efficiency concerns.

Note that FLEXIBASE will perform the capacity change by the fastest method compatible with the disc resources available.

If there is insufficient free disc space (or the disc space is too fragmented) for FLEXIBASE to build its work files on disc, the following steps are taken:

The operator is first informed of the exact disc space shortage and asked to either make some more space available for a re-try, or to permit the software to use a tape work file.

If a tape work file is selected and the software writes the old data set to tape and purges the old set, but there is still insufficient disc space to build the new set, then the operator is again informed of the space shortage and asked to either make more space available for a re-try or to permit abandonment of the capacity change. If abandonment is requested then the old set is recovered from the tape file.

It is possible to avoid the operator interaction in these cases by setting a JCW named FLEXTAPE before the run. If the JCW is assigned a value of zero then tape work files will never be used (in this case the capacity change will be abandoned if insufficient disc space is present and the database will be left intact). Assigning a value of 1 to the JCW will invoke the use of tape work files (when necessary) without asking permission of the operator.

THIRD PARTY INDEXES

If FLEXIBASE detects a TPI index on the dataset, then it will automatically call the appropriate subroutines (either Omnidex or SUPERDEX) to rebuild the index. The following message will be displayed:

```
Database name >> sales
TPI is enabled - calls will be made as necessary to re-synchronize indexes
.
.
Calling TPI product to re-index set....
Re-indexing completed OK
```

Should an error occur for any reason, or should the TPI product be unable to service the request, then FLEXIBASE will warn you of the fact, as follows:

```
Calling TPI product to re-index set....
Error in re-indexing set - User attention required      -1
```

You should then use the indexing tool directly to resynchronize the indexes, after the FLEXIBASE run has completed.

To disable the calls to the third party product because the version is inappropriate or for some other reason, set the following JCW before the run:

```
:SETJCW FXNOTPI=1
```

Note that automatic reindexing will only be available if the database has been enabled for TPI in DBUTIL.

Note that reindexing will only be necessary if the capacity is changed such that it falls below the existing High Water Mark.

B-TREE INDEX

Where a master with a B-Tree index is having its capacity increased, the following message will show:

```
Re-Building B-Tree to new size……
```

SECTION 7 ℂ AUTOMATIC CAPACITY FUNCTION

In addition to the Capacity Change facility described in section 6, the FLEXIBASE software also contains an automatic capacity management function. The purpose of this function is to allow capacities of sets to be automatically monitored and changed when necessary according to parameters initially set by the user. The parameters can be simple percentages of the set capacity or a complex formula. The normal procedure would be to run this function to build a job file containing the parameters, and then stream the job on a regular basis (e.g. daily or weekly). The job file can control some or all sets in one or more databases. Alternatively separate job files can control different groups of sets in one database.

```
Requirements & Pre-conditions:
•   Ensure that a back-up copy of the database is available.
•   The user must be logged on as the database creator.
•   Exclusive access to the database (mode 3) is required.
•   MPE read, write, and save access to both the log-on group and the group in which
    the database resides is required.
```

The following is the dialog between the user and the FLEXIBASE software:

```
:RUN FLEXIBAS.PUB.HQ
```

Select option 6 from the menu.

```
Database name >>
```

Enter the name of the database in which the capacity management is required. The user must be the creator of the database and the software requires exclusive access to the database (except when setting up a job file).

```
Set name (or @, @M, @A, @D) >>
```

Enter an individual dataset name, or @ for ALL sets, or @M for all Manual Master sets, or @D for all Detail sets, or @A for all Automatic Master sets.

```
INCREASE CAPACITY TRIGGER: Enter MAX % FULL
allowed (or MIN CAPACITY formula) >>
```

Enter either the maximum percentage full to be tolerated for the specified sets before the capacity is to be increased, e.g. 80% (the figure entered must be in the range 50 to 99%); or enter a formula which expresses the minimum capacity allowed. If the set has a capacity lower than the specified minimum capacity, then the capacity will be increased. See the end of this section for the valid syntax for the entry of a formula.

```
TARGET IF CHANGING: Enter TARGET % FULL (or NEW CAPACITY formula) >>
```

When a set capacity is increased, or decreased, it is done so as to make the set percentage full or new capacity equal to this target. If a percentage is entered, it must be less than the previously entered INCREASE CAPACITY TRIGGER figure.

```
DECREASE CAPACITY TRIGGER: Enter MIN % FULL
allowed (or MAX CAPACITY formula) >>
```

Enter either the minimum percentage full to be tolerated for the specified sets before the capacity is to be decreased, e.g. 40% (the figure must be less than the target percentage, and the minimum is effectively disabled if zero is entered); or enter a formula which expresses the maximum capacity allowed. If the set has a higher capacity than that specified, then the capacity will be decreased. See the end of this Section for the valid syntax for the entry of a formula.

Explanation and Notes on the above three prompts:

If you simply want to keep the percentage full of the set within a given range, say 50% to 85% full, then just enter three percentage numbers, in descending order, in response to the three prompts, e.g. 85%, 70%, 50%. The result will be that as soon as the percentage full goes outside those limits, that is above 85% or below 50%, then the next Automatic Capacity Run will change the capacity back to the middle figure i.e. 70% in this example.

The above is the straightforward way of maintaining capacities for most cases. In some instances however, because of particular circumstances, it may be appropriate to use the more sophisticated and flexible method of capacity formulas.

For example:

a) If you wanted to control the capacity by number, rather than by percentage, you could enter

ENT+1000, ENT+3000, ENT+5000

In this case, if the capacity became less than 1000 over the number of entries, or more than 5000 over the number of entries, then the capacity would be changed back to the number of entries plus 3000.

b) Suppose you have a detail set to which are added a large number of entries during the day, but from which most of the entries are processed and deleted by the evening. It would not be possible to control the capacity of the set using percentages full, since the set is fairly empty when the Automatic Capacity Change is run. However, the High Water Mark "remembers" the maximum number of entries which the set has contained, and can therefore be used to control the capacity.

E.g. HWM*1.25, HWM*1.5, HWM*2

In this case, if the High Water Mark ever changes such that the capacity is less than one and a quarter times the HWM, or more than twice the HWM, then the capacity would be changed back to one and a half times the HWM.

c) Perhaps in a sales analysis system, a set will contain one entry for each product for each month, building up as the year progresses.

If we use

ENT(PRODUCTS)*MON,   ENT(PRODUCTS)*MON, ENT(PRODUCTS)*MON

then the Automatic Capacity Run will always adjust the capacity of the product analysis set to the appropriate figure, i.e. the number of entries in the products set multiplied by the month number.

See the end of this section for full details of the syntax rules and operands which can be used in formulas.

After the above prompts for the capacity change parameters, FLEXIBASE reviews the set entries, and the capacity change function is invoked when required if running in session mode. Sets that have the capacities changed will be listed.

If you are short of disc space or wish to control use of tape work files then refer to the notes in the Section entitled "Capacity Function" for more information.

Note: A Job Control Word (JCW) named FLEXAUTO is set with a value of 1 if any sets actually have capacity changes actioned in the run, else it is not set. This enables the user to take different actions in a jobstream (e.g. DELETE STDLIST or TELL OPERATOR) depending on whether or not capacity changes have actually taken place.

SYNTAX RULES FOR CAPACITY FORMULAE

The algebraic operators of **+**, **-**, *****, and **/** for add, subtract, multiply and divide respectively, are allowed on the following types of operands:

`ENT(setname)`	means the number of entries in the named set.
`CAP(setname)`	means the capacity of the named set.
`HWM(setname)`	means the High Water Mark of the named set.
`9999.99`	means a numeric literal.
`MON`	means the current month number from the system date.
`JCW`	means the value of the FXCAPJCW Job Control Word.

The use of ENT, CAP or HWM without a set name in brackets afterwards implies the current set. Spaces within the formula are ignored. Algebraic brackets and commas within numbers are not allowed. The High Water Mark (HWM) of a master set is considered to be the same as its capacity. The percent sign is not allowed within a formula (although a percentage is a valid alternative to a formula, when it represents the percentage set full). The three responses to the prompts for INCREASE, TARGET and DECREASE can be a mixture of percentages and formulae.

Examples of valid entries are:

```
>> ENT(TRANSACTIONS) * 2
>> CAP * 1.25
>> HWM + 1000
>> ENT(PRODUCTS) * 1.2 + ENT(AGENTS) * 1.25 + 3000
>> CAP(CUSTOMERS) – 5000 + ENT(MARKETS) / 2
>> 25010
>> ENT(EMPLOYEES) * MON + 50
>> ENT + JCW
```

HIGH WATER MARK

The High Water Mark of a detail set is the maximum number of entries that the set has ever contained since it was created (or since the last REPACK or TRANSMUTE run on it). It is equal to the current number of entries plus the current number of deleted entries (the latter can be obtained from the DIAGNOSE statistics). A warning is issued if the capacity is to be reduced below this mark since the run time will be longer in that case.

THIRD PARTY INDEXES

If FLEXIBASE detects a TPI index on the dataset, then it will automatically call the appropriate subroutines (either Omnidex or SUPERDEX) to rebuild the index. The following message will be displayed:

```
Database name >> sales
TPI is enabled - calls will be made as necessary to re-synchronize indexes
.
.
Calling TPI product to re-index set....
Re-indexing completed OK
```

Should an error occur for any reason, or should the TPI product be unable to service the request, then FLEXIBASE will warn you of the fact, as follows:

```
Calling TPI product to re-index set....
Error in re-indexing set - User attention required      -1
```

You should then use the indexing tool directly to resynchronize the indexes, after the FLEXIBASE run has completed.

To disable the calls to the third party product because the version is inappropriate or for some other reason, set the following JCW before the run:

```
:SETJCW FXNOTPI=1
```

Note that automatic reindexing will only be available if the database has been enabled for TPI in DBUTIL.

Note that reindexing will only be necessary if the capacity is changed such that it falls below the existing High Water Mark.

B-TREE INDEX

Where a master with a B-Tree index is having its capacity increased, the following message will show:

```
Re-Building B-Tree to new size……
```

SUPPORT FOR CONTROL DATASETS

Many applications use one or several datasets with a capacity of one to hold single control records. Because such datasets are by their nature 100% full, they would necessarily be updated by Automatic Capacity Change. To avoid this happening, the Automatic Capacity function will not modify datasets with a capacity of one. Should you need to modify one, use the CAPACITY function to increase the capacity (see section on CAPACITY FUNCTION).

SECTION 8 ℂ REPACK FUNCTION

8.1 OVERVIEW

The purpose of this module is to resequence entries in a detail set so that the physical sequence of entries corresponds to the logical sequence of entries. The result is increased performance when accessing the set using chained access (DBGET mode 5) on the primary path, since each chain of entries can be retrieved from disc with the minimum number of physical accesses to disc.

The resequencing is achieved by an algorithm which avoids the need to unload and reload the set. The speed of resequencing is not affected by the presence of sorted paths, and is very efficient at dealing with multiple paths. The REPACK run time is also very much reduced when the set is marginally messy, so users are recommended to use the REPACK function on a regular basis.

```
Requirements & Pre-conditions:
•    Ensure that a backup copy of the database is available.
•    Exclusive access to the database is required.
•    The user must be logged on as the database creator.
•    MPE read, write and save access to the database group is required.
```

8.2 USER DIALOG

The following is the dialog between the user and the FLEXIBASE software:

```
:RUN FLEXIBAS.PUB.HQ
```

Select option 7 from the menu.

```
Database name >>
```

Enter the name of the database containing the set(s) to be repacked. The user must be the creator of the database, and the software requires exclusive access to the database (except when setting up a job file).

```
Set name (or @ for all sets) >>
```

Enter the name of the detail set to be repacked, or enter @ to repack all detail sets in the database.

```
The following Disc Workspace is Required to REPACK by :
the FASTEST method (ON DISC)    78260 sectors(19.1 Megabytes)
the FAST method (USING TAPE)     6193 sectors( 1.5 Megabytes)
the SLOWER method (USING TAPE)      0 sectors( 0.0 Megabytes)

Assessing currently available discspace........

Proceeding NOW would be by the FASTEST method (ON DISC)

Do you want to proceed - Enter Y or N >>
```

The software calculates the disc workspace required to perform the REPACK on this set by each of the algorithms available and quotes the figures calculated. It then assesses how much disc workspace is currently available, in order to inform the user which algorithm would be invoked if the run proceeds at this time. (See the discussion of the algorithms later in this section). Finally the software asks for confirmation before proceeding to make the physical reorganisation.

```
EMPLOYEES set being processed.....

<><><> SET HAS BEEN REPACKED <><><>
```

This message confirms successful completion of the operation.

8.3 REPACK ALGORITHMS

The fastest method of REPACKING, and therefore, in throughput terms the most desirable method, requires an amount of disc workspace averaging 1.1 times the size of the set. If this amount of disc space is readily available then the software can proceed using the fastest method.

For very large sets however, or in cases where free disc space is at a premium, then it is possible for the software to employ a different method which uses a TAPE device for most, but not all, of its workspace. Using this method the amount of disc space required is reduced to an average of 10% of that required for the fastest method, but the run-time is increased by about 40%. (The exact disc requirement and run time differences depend on the number of paths into the detail set). Thus this method is invaluable for large sets, or in situations of disc space shortage, since it enables the sets to be REPACKED, which otherwise might not be possible, but with only a modest run-time premium.

Finally, in situations of dire disc space shortage, the software can revert to a slower algorithm employing an unload and reload of the entries via TAPE. This method requires virtually no disc work space, but the run-time is significantly greater than the above two methods. It is likely that this method will be used only rarely.

The above combination of three methods depending on available free disc space thus gives the user the best possible combination of speed and capability. The software itself will automatically choose the fastest method commensurate with the disc space available at the time of the run, though the user is informed of space requirements and the method chosen before being asked to confirm that the run can proceed.

The following are figures from actual test runs:

		DISC REQUIRED	RUN-TIME
SET with 1 path	FASTEST method (ON DISC)	16765 sectors	650 seconds
	FAST method (USING TAPE)	31 sectors	939 seconds
	SLOWER method (USING TAPE)	0 sectors	7802 seconds
SET with 2 paths	FASTEST method (ON DISC)	20325 sectors	669 seconds
	FAST method (USING TAPE)	1281 sectors	1070 seconds
	SLOWER method (USING TAPE	0 sectors	2503 seconds
SET with 3 paths	FASTEST method (ON DISC)	35465 sectors	2986 seconds
	FAST method (USING TAPE)	10603 sectors	3294 seconds
	SLOWER method (USING TAPE)	0 sectors	18047 seconds

8.4 TAPE HANDLING

If the user wishes to change the default of "TAPE" to another serial device, e.g. cartridge tape, then a file equation should be entered preceding the RUN command, which equates the file name FXUNLOAD to the selected device. e.g. :FILE FXUNLOAD;DEV=CTAPE

Normally, if the software requires to use TAPE then the operator console will be informed and asked to confirm whether to proceed using TAPE or whether to perform a safe abort leaving the set in question untouched. This operator decision can be pre-empted by setting a JCW named FLEXTAPE prior to the RUN command.

:SETJCW FLEXTAPE=0   This will prevent tape being used. If insufficient disc workspace is available for the FASTEST method then the REPACK will not be performed.

:SETJCW FLEXTAPE=1   This will allow the software to proceed to use TAPE (if it is required) without permission of the operator. This will only be useful in an unattended situation if a reel is left on-line, and one reel is sufficient for the run. (The number of reels required is roughly equivalent to that required for a STORE of the set.)

8.5 OTHER POINTS

Note that the original chronological sequence of all paths is maintained during a REPACK run by both the FASTEST method (ON DISC) and by the FAST method (USING TAPE) and by the SLOWER method (USING TAPE). i.e. the entries remain logically (though not physically) in the same sequence as they were added to the set. This is important to some applications e.g. HPDESK.

Users of MPE/iX systems REPACKing large sets which exceed twice the main memory size, can gain a performance improvement by setting a JCW prior to the set being repacked which changes I/O optimisation :SETJCW FXBIGSET=1. Whilst the setting of this JCW can reduce run-times on very large sets by up to 25%, it should not be set during smaller set REPACKS since it will degrade performance by a corresponding amount.

8.6 THIRD PARTY INDEXES

If FLEXIBASE detects a TPI index on the dataset, then it will automatically call the appropriate subroutines (either Omnidex or SUPERDEX) to rebuild the index. The following message will be displayed:

```
Database name >> sales
TPI is enabled - calls will be made as necessary to re-synchronize indexes
.
.
Calling TPI product to re-index set....
Re-indexing completed OK
```

Should an error occur for any reason, or should the TPI product be unable to service the request, then FLEXIBASE will warn you of the fact, as follows:

```
Calling TPI product to re-index set....
Error in re-indexing set - User attention required      -1
```

You should then use the indexing tool directly to resynchronize the indexes, after the FLEXIBASE run has completed.

To disable the calls to the third party product because the version is inappropriate or for some other reason, set the following JCW before the run:

```
:SETJCW FXNOTPI=1
```

Note that automatic reindexing will only be available if the database has been enabled for TPI in DBUTIL.

SECTION 9 ℂ REPAIR FUNCTION

The purpose of this module is to ensure correct chaining of all entries in a detail dataset, so that DBGET modes 5 and 6 will function correctly. This module is only used if a broken chain has resulted from an untimely abort of database processing, e.g. because of a system failure or similar problem, and will normally have been diagnosed by an IMAGE error message or by running such programs as DBCHECK or DIAGNOSE. This module will function correctly whether or not any broken chains are actually present so it can be used as a precautionary measure when the chaining is merely suspect.

The rechaining is achieved by unloading (in serial sequence) all entries from the specified set to a serial file (disc or tape), erasing the set, and then reloading all entries back into the set. The user simply initiates the run and all three phases are then automatically performed.

```
Requirements & Pre-conditions:
•   Ensure that a backup copy of the database is available.
•   Exclusive access to the database is required.
•   The user must be logged on as the database creator.
•   MPE read, write, and save access to the database group is required.
```

The following is the dialog between the user and the FLEXIBASE software:

```
:RUN FLEXIBAS.PUB.HQ
```

Select option 8 from the menu.

```
Database name >>
```

Enter the name of the database containing the set to be repaired. The user must be the creator of the database, and the software requires exclusive access to the database (except when setting up a job file).

```
Set name >>
```

Enter the name of the set to be repaired.

```
<><><> REPAIRS ARE COMPLETED <><><>
```

This message confirms successful completion of the operation.

The software will attempt to create a file on disc sufficient to contain all the unloaded entries. If disc space is not available then the software will open a file on device "TAPE" and request a volume be mounted.

If the user wishes to change the default of "TAPE" to another serial device, e.g. cartridge tape, then a file equation should be entered preceeding the FLEXIBASE run command which equates the file name FXUNLOAD to the selected device, e.g. :FILE FXUNLOAD;DEV=CTAPE.

If the user, for reasons of disc space management, does not want the software to build a disc file, then disc availability should be restricted by use of the "ALTGROUP;FILES=" command until the first volume has been requested.

Third Party Indexes

If FLEXIBASE detects a TPI index on the dataset, then it will automatically call the appropriate subroutines (either Omnidex or SUPERDEX) to rebuild the index. The following message will be displayed:

```
Database name >> sales
```

```
TPI is enabled - calls will be made as necessary to re-synchronize indexes
.
.
Calling TPI product to re-index set....
Re-indexing completed OK
```

Should an error occur for any reason, or should the TPI product be unable to service the request, then FLEXIBASE will warn you of the fact, as follows:

```
Calling TPI product to re-index set....
Error in re-indexing set - User attention required      -1
```

You should then use the indexing tool directly to resynchronize the indexes, after the FLEXIBASE run has completed.

To disable the calls to the third party product because the version is inappropriate or for some other reason, set the following JCW before the run:

```
:SETJCW FXNOTPI=1
```

Note that automatic reindexing will only be available if the database has been enabled for TPI in DBUTIL.

SECTION 10 C TRANSPOSE FUNCTION

The purpose of this module is to allow values of data-items to be globally changed throughout a database. The changes can be effected whether the item occurs as a search-item, sort-item or ordinary item, or a mixture of all three in different datasets.

For example: Suppose a database contains a master set of CUSTOMERS of which the search-item is CUSTOMER CODE. A detail set of TRANSACTIONS is chained to the CUSTOMERS set by that search-item. If the CUSTOMER CODE is required to be amended then TRANSPOSE can amend the value of the single master entry and all its related detail entries in one operation. Amends of search-items and sort-items are not possible using QUERY, so the general procedures of TRANSPOSE can save the need to write one-off programs to effect difficult or wide-ranging amendments.

TRANSPOSE is designed to efficiently process multiple values for multiple data-items in one pass of each set where a serial update is required, though direct and chained-access are used wherever possible.

```
Requirements & Pre-conditions:
•    Ensure that a backup copy of the database is available.
•    Exclusive access to the database is required.
•    The user must be logged-on as the database creator.
```

The following is the dialog between the user and the FLEXIBASE software:

    :RUN FLEXIBAS.PUB.HQ

Select option 9 from the menu.

```
Database name >>
```

Enter the name of the database in which the amendments are to be made. The user must be the CREATOR of the database, and the software requires exclusive access to the database (except when setting up a job file).

```
If new search-item value already exists in master
set, should detail chains be MERGED under the
existing master entry (Y/N) >>
```

Normally if the user enters a new data-value for a search-item which already exists, then a "Duplicate value" error message is emitted at the time of data-entry, and that amendment will be ignored. This will be the case if the user enters "N" at this point.

If however it is required to delete the old data-value entry, and merge all its dependant detail chains under the existing new data-value entry, then the user should enter "Y" at this point to trigger this "MERGE" mode of operation.

```
Do you want to INCLUDE ONLY certain sets in serial searches (Y/N) >>
```

If the user does not wish to limit the processing to certain sets only then enter "N" at this point. If however the user enters "Y" at this point then the following repeated prompt will allow specification of which sets are to be INCLUDED in the serial reading process. (Note that TRANSPOSE will in any case always read master sets and dependant detail chains in which the item to be amended is a search-item - this being the critical but least time-consuming aspect of modification. The restriction of the scope of amendment is only applied to sets which contain the item not as a search-item.)

```
Setname to be INCLUDED in serial search [press return to end] >>
```

Enter the set-name of each set which is to be INCLUDED in the scope of amendments. After all relevant set-names have been entered then just press return to terminate the list.

```
Do you want to EXCLUDE certain sets from the serial searches (Y/N) >>
```

If the user does not wish to exclude certain sets from the processing then enter "N" at this point. If however the user enters "Y" at this point then the following repeated prompt will allow specification of which sets are to be EXCLUDED from the serial reading process. (Note that TRANSPOSE will in any case always read master sets and dependant detail chains in which the item to be amended is a search-item - this being the critical but least time consuming aspect of modification. The restriction of the scope of amendment is only applied to sets which contain the item not as a search-item.)

```
Setname to be EXCLUDED from serial search [press return to end] >>
```

Enter the set-name of each set which is to be EXCLUDED from the scope of amendments. After all relevant set-names have been entered then just press return to terminate the list.

```
Data-item name >>
```

Enter the name of a data-item which is to have values amended (or just press return to end the input of items and values.) The data-item may be of any data-type or length. A subitem of a data-item may be indicated by including its index in square brackets after the name e.g. SALES-MONTH[4] . If you omit the index from an item which contains sub-items, then the first sub-item is assumed as default.

```
Old data value >>
```

Enter the value of data which, if it occurs in the above data-item, is to be amended. The data is automatically validated to be of an appropriate type and length for the data-item. Character-fields are automatically padded with spaces to the maximum length. A maximum of 60 characters may be entered. (Or just press return to terminate the entry of pairs of data-values for this data-item.)

Leading spaces will be stripped unless the data value is enclosed in quotes (either double or single quotes). A data value which is enclosed within quotes may itself contain quotes.

```
New data value >>
```

Enter the value of data to which the above old data-value is to be amended. The data is automatically validated to be of an appropriate type and length for the data-item. Character-fields are automatically padded with spaces to the maximum length. A maximum of 60 characters may be entered. (Or just press return to cancel this pair of old and new data-values.)

Leading spaces will be stripped unless the data value is enclosed in quotes (either double or single quotes). A data value which is enclosed within quotes may itself contain quotes.

The prompts for data-item names with pairs of associated values will be repeated until the user answers the data-item prompt by pressing return. At this point the software will ask for confirmation before making any physical modification to the database.

```
Do you want to proceed - Enter Y or N >>
```

Answer "Y" to carry out modifications. If you answer "N" the database will remain in its original state.

```
Example Run Report:

Set              Item            Amends Old Data        New Data

CUSTOMERS        ACCOUNT-CODE         1 SMITH           ABC CORP
 TRANSACTIONS                        13
 SALES-ANALYSIS                      22
CUSTOMERS                             1 PQY CORP        ACME
 TRANSACTIONS                         3
 SALES-ANALYSIS                      45


Serial read of PRODUCT-ANALYSIS      37% Complete
PRODUCT-ANALYSIS ACCOUNT-CODE       572 SMITH           ABC CORP
```

```
                                39 PQY CORP        ACME
```

All amendments are reported. The first block above shows the ACCOUNT-CODE search-item in the CUSTOMERS set being amended for two values (SMITH & PQY CORP), and in each case chains of entries in the detail sets TRANSACTIONS and SALES-ANALYSIS also being updated. The indentation of the latter two set names indicates that they are detail sets connected by paths. This is followed by a serial read of set PRODUCT-ANALYSIS with progress reporting, to find values of ACCOUNT-CODE (which is not a search-item in that set).

```
    <><><><> ALL AMENDMENTS COMPLETED <><><><>
```

This message confirms successful completion of all amendments.

ITEMS of DATA-TYPE P in TRANSPOSE

If the name of a data-item is entered which has a data-type of P, then an additional question is asked by the software:

```
Data-type P : formatted for Unsigned COBOL, Signed COBOL, or RPG ?

Enter U,S or R >>
```

Positive Signed values from COBOL are stored in a subtly different bit pattern than those from other language sources. (TurboImage Reference Manual 3-7) The user response is used, if necessary, to identify the appropriate bit pattern on reading, and to format correctly when writing. The question is asked only on the first user reference to a P-type item, and the response is applied to all P-type items in that run.

SECTION 11 ℂ CREATE FUNCTION

11.1 PURPOSE AND USER DIALOG

The CREATE function can be used to create a new, cut-down version of a database. It copies a database but selects a limited number of records. Records can be selected on a random basis (e.g. 20% of all records) or on the basis of specific item values (or range of values).

All associated records in other data sets can be copied so that the new data base can be a logical subset of the old one. This function is particularly useful for creating small test databases but it also has other applications in the manipulation of operational databases.

The new database is created in the log-on group during the run, if it does not already exist (see below). The user must have the appropriate access to the group and account of the main source database to be able to DBOPEN the database for read access. The newly created sample database will have all flags (logging etc.) set to their defaults. Remote databases can also be sampled from, but in this case direct access to the source root file is not possible and so passwords and read/write lists are set to default values.

The primary purpose of the CREATE module is to create small sample test databases. By enabling sampling of all logically connected entries a consistent subset of data entries can be created. Such a subset could provide test data for program development in a very short time. Such sampling can be done while other users are accessing the source database and so test data can be created as and when required. The facility for consistent subsets of data also enables the application of creating logical sub-databases. For example, if an accounts database contains data for more than one company it may be possible to identify only those entries relating to one of the companies and select them into another, smaller, database.

```
Requirements & Pre-conditions:
•   The user must be able to open the source database in mode 5 or 6.
•   The user must supply a password which gives complete read access to ALL sets and
    items in the source database.
•   The user must have read, write, and save access to the log-on group, or to the
    group in which the destination database resides.
```

The following is the dialog between the user and the FLEXIBASE software:

```
:RUN FLEXIBAS.PUB.HQ
```

Select option 10 from the menu.

```
Do you want to run Interactively now or create a streamed Job file?
     Enter I or J >>
```

The actual creation of the sample database can either be performed in session mode or defined and run as a job.

```
Source Database Name >>
```

Enter the name of any existing database from which you wish to transport entries into the Sample Database. The Database Name may be fully qualified and you must supply a password which will give complete read access to all parts of the source database. File equations and remote databases are allowed. Note that CREATE will fully qualify the source database name before attempting to open it, thus if a file equation is required it must relate to the fully qualified database name. For example, if the program is run while the user is logged into the PUB group of the HQ account and the source database name TESTDB is entered then CREATE will open the database TESTDB.PUB.HQ. If the user back-references this to another database he must enter the file equation :FILE TESTDB.PUB.HQ=<<other database name>>. Note: As long as the file name on the left hand side of the file equation is fully qualified, it can be any valid MPE file name (e.g. X.Y.Z or DB.GROUP.ACCT).

```
Password >>
```

Enter relevant password. If the source database is a remote database then this password will be the only known password from the source database, all others will be defaulted. Thus, this password *must* give full read access to the entire source database. In any event the password should give read access to all of the source database to enable complete copying of entries, to enable the correct set capacities to be used and to avoid inexact copying of the source database.

```
Destination Database Name >>
```

Enter a valid database name for the new Destination Sample Database. This database will be created as an exact copy of the Source database but with reduced Capacities (except in the case of a remote source database - see above), and will be created in the logon group (the database name may not be qualified). This database may already exist (in the case where a previous CREATE run was performed for example). However, if the database already exists then unless the structures of the source and destination bases are completely identical the software will give unpredictable results and may abort.

```
Enter Capacity Percentage or Formula >>
```

Enter the percentage of the Source Database Set Capacities to be used in the Destination Sample Database. This may be expressed as a percentage of the Source Capacities or as a percentage of the number of entries in the Source Database. Specifying a number on it's own instructs CREATE to calculate the Destination Database's Capacities as a percentage of the Source Database's Capacities. Thus, entering 10 will cause the Destination Capacities to be one tenth (10%) of the Source Capacities. The value entered must be greater than zero and less than or equal to 100. To use the Source Database's entry counts as the basis for the calculation enter "ENT*" followed by the percentage required. Thus, entering ENT*10 instructs CREATE to calculate the Destination Database's Capacities as 10% of the Source Database's Entry Counts. The value entered must be greater than zero and less than or equal to 200. In this way the Destination Database may be created with Capacities up to twice that of the number of entries in the Source Database.

Having specified the overall capacity percentage the newly-calculated capacities of the sample database are now displayed. Individual set's capacities may be altered as required. Thus you may override the automatic calculations if required.

```
The Sample Database will have the following Capacities:

        Dataset Name   Type   Capacity     Dataset Name   Type   Capacity
        CLOCK-NOS      A      157          DEPARTMENTS    M      23
        BANKS          M       37          DUMMY          M       1
        EMP-DATA       M      157
Do you want to amend any of these Capacities (Enter Y or N) >>
```

Enter Y to amend individual sets' capacities by specifying the required capacities.

```
Dataset Name to amend >>
```

Enter the dataset name whose capacity you wish to modify from the value displayed above.

```
Capacity Requested >>
```

Enter the requested capacity for this set, e.g. 127.

```
Creating Sample Database.....(please wait).....
```

If in session mode then the new sample database will be created at this point. In job mode the database is not created and any future validation is done against the Source Database. If the Source database is located on a remote machine then at this point the following lines are displayed on $STDLIST:

```
N.B As SOURCE DATABASE is REMOTE, default passwords of 5 READER & 10 <<pass>>
--- and default read/write lists (5/10) will be inserted in the sample database.
```

In the case of a remote source database, passwords are not available and thus the item and set read/write lists are set to defaults of (5/10). Password level 5 is defaulted to the password READER, and password level 10 is set to the password used to open the source database previously. If the remote source database was opened with the ";" password then a password 10 WRITER is used.

In order for CREATE to navigate through the database structure when copying entries it also temporarily changes all Automatic Master Sets in the destination database into Manual Master Sets. At the end of the run they are reset back to Automatics. The user should NOT, however, break and abort a CREATE run as the Automatic sets will not have been changed back to Manuals. This is very important.

```
Copy by Item Values or by Random Selection (Enter V or S) >>
```

Having created the Sample Database there are two ways of transporting entries across from the Source Database:

1.   All entries related to specific data-item values in a specified set can be selected (i.e. all entries in that set plus entries in other sets associated by paths to that set and their associated chain heads or detail chains).

2.   Randomly fill master sets up to 85% full, and attempt to fill detail sets only where chain heads exist.

Of the two methods 1 will result in a much more consistent Sample Database than 2 because logically connected entries can be transported across, as can subsets of data. Method 2 will not guarantee consistency, but gives a fast, general and random filling of the database.

Enter V to transport entries across according to particular data-item values.

Enter S to randomly select entries from the Source Database.

11.2 COPYING BY VALUE

This option allows you to transfer individual entries across from the Source Database to the Sample Database. Specify a "key" dataset and particular values of any item in that set and the matching entries will be transported across, along with as many associated detail/master entries as possible. Up to three data-item value conditions may be entered, and the wildcard character "@" may be used to refine those entries to be transported.

It is important to understand the process which CREATE uses to transport entries across. Starting with the "key" dataset specified CREATE will identify the entry or entries in that set matching the data-item value conditions entered. If this set is a detail set then CREATE will have to also transport entries in those master sets which have paths into the detail set (to provide the correct chain heads). These master sets may have other detail sets associated with them by paths - the entries in these detail sets for the relevant values will also be transported. These details may themselves require master set entries to be transported to provide chain heads, and so the whole process repeats itself until all paths are exhausted, or sets fill up. Provided the Destination Set capacities are large enough ALL logically connected data set entries will be transported across. There is no practical limit to the number of sets involved, nor the number of entries copied.

The dialog is as follows:

```
Dataset Name >>
```

Enter the name of any of the Datasets in the Source Database from which you want to transport entries with matching data-item values. Any Manual Master, Automatic Master or Detail Sets associated to this set by paths will have their associated chains or chain heads transported also (along with their associated entries). Any set may be specified, even Automatic Masters (as these are temporarily modifed to become Manual Masters for the purposes of the CREATE run). The set should be selected carefully so as to ensure as complete sampling as possible.

```
Enter Data Value Condition(s) >>
```

The user may now specify up to three data item value conditions which will be used to select matching data set entries. Each condition is composed of a data item name, a relational operator and a value. If more than one condition is entered then they must be separated by an "AND" or an "OR".

Thus, the general syntax can be represented as follows:

> ITEM RELOP VALUE
> [AND/OR ITEM RELOP VALUE
> [AND/OR ITEM RELOP VALUE]]

where:

ITEMNAME represents the name of any of the Data-items in the specified dataset, and may include the specification of a specific sub item number (enclosed by parentheses).

RELOP is one of the allowed relational operators ">", "<", "=", "<>", ">=" or "<=". Spaces may be entered around relational operators, but are not mandatory and are ignored.

VALUE is a relevant data item value for the specified data item. The wildcard character "@" may be used to specify ranges of values.

AND/OR is either the words AND or OR. These are mandatory if more than one condition is entered.

For example, to select all entries which have a data value of 20000 for the ACCOUNT-NO item in the specified set (assuming ACCOUNT-NO to be a valid item within the specified set) the user would enter:

```
ACCOUNT-NO=20000
```

This may also be entered with spaces, eg ACCOUNT-NO = 20000; or with quotes, eg ACCOUNT-NO="20000". The spaces and quotes are ignored, except in the instance when the quotes may be used to specify preceeding spaces in a data item value.   Thus, REFERENCE=" A" is different to REFERENCE="A".

Selecting on multiple (up to three) data value conditions is a powerful way of refining the specific entries required. Each condition must be separated by AND or OR, further expanding the possibilities.

Thus, to sample those entries with a market-code of IT and an industry-code of BRICK the user would enter:

```
MARKET=IT AND INDUSTRY=BRICK
```

Note that the data item names, and AND/OR, may be in upper or lower case (or even a mixture of the two) but the data item values must be in the appropriate case to be matched correctly.

The following are valid examples of data value conditions (assuming that the items specified exist in the specified set:

```
NAME > "A" and NAME < "B"
AMOUNT >= 0 AND amount <= 200
COMPONENT(12)<>" "
MARKET=IT or Market=UK or market="FR"
INDUSTRY<>"SAN" AND INDUSTRY>"REFR"
ADDRESS(2)>XXXXXX
```

The wildcard character "@" may be used to further refine the selection criteria. Thus entering STOCK-NO="PAP@" will select only entries with a STOCK-NO whose first three characters are PAP. Each of the three potential conditions may include the wildcard character - but it may only be associated with the "="

relational operator. Thus SUPPLIER > A@ is invalid as it does not use the "=" operator. Examples of it's valid use are:

```
SUPPLIER=COMP@
ACCOUNT-NO = 2@ AND MARKET <> UK
BALANCE>0 or NAME=ICI@
DATE=@91
```

The software now attempts to transport entries from this set and any sets connected by paths to this set (and any sets connected to them by paths, and so on). For example, suppose the requested set is a Manual Master and one condition is entered (eg Search-Item=123456). This Master set has two Detail Sets, each of which has one other Master Set (which are not connected to any other sets). The result would be: one entry transported from the Manual Master Set (assuming the entry exists with the requested value); both Detail Set chains for this Search Item value (if any, and as long as the Detail Sets do not become full); and any entries in the other two Master Sets required to provide chain heads for the Detail Set chains. The output would look like the following:

```
Copying entries by value.....
MASTER-1         set copied .....        1 entries   ( 5% full)
MASTER-2         set copied .....        2 entries   ( 7% full)
MASTER-3         set copied .....        5 entries   ( 6% full)
DETAIL-1         set copied .....       15 entries   ( 12% full)
DETAIL-2         set copied .....       55 entries   ( 22% full)
```

11.3 PROGRESS REPORTING

In order to keep the user aware of how far through a sampling CREATE has processed it displays progress messages on $STDLIST. These messages are to be used as approximate guides only as it is not possible to accurately estimate the time required or number of entries which will be selected before they are actually sampled. The name of either the specified data set, or one of it's associated detail sets, will be displayed along with the number of entries which have been transported across for that set.

A percentage figure is shown which attempts to represent the percent complete for this dataset, but may not always be accurate - they are meant as a guide only. These messages are displayed every 10 seconds (unless specified otherwise - see Appendix J). In addition a file FXPROGRn (where n is a single digit 1-9) is created which contains the same information. Normally this file will be called FXPROGR1, but if another user is already using CREATE or DIAGNOSE in the same logon group then the file will be FXPROGR2 or 3 etc.

11.4 COPYING BY RANDOM SELECTION

This option allows you to transport randomly selected entries across from the Source Database to the Sample Database. Specify a specific dataset, or @ to transport entries across from all sets. As many Master entries as possible will be transported up to 85% of the capacity of the set in the Destination Database. Detail Set Entries will only be transported if chain heads exist in their corresponding Master Sets (again up to a maximum of approximately 85% of the Destination Set Capacity). For this option CREATE reverts all Automatic masters to their native state, and thus these are transported automatically by virtue of their detail sets being transported.

```
Dataset Name or @ for all Sets >>
```

Enter the set name of any Manual Master or Detail set in the Source Database from which you wish to transport entries across into the Destination Sample Database. Enter @ to randomly select entries from all master sets along with their associated detail chains.

The output will look like the following:

```
CLOCK-NOS        set ℂ  automatic master
```

```
DEPARTMENTS      set copying .....     13 entries ( 87% full)
BANKS            set copying .....     17 entries ( 86% full)
DUMMY            set copying .....      0 entries ( 0% full)
EMPLOYEES        set copying .....     71 entries ( 86% full)
LOCATIONS        set copying .....      3 entries ( 57% full)
```

Progress messages are provided to indicate the progress of the sampling.

When no further copying of entries is required from this particular Source database (signified by pressing return at the question "Copy by Item Values or Random Selection (Enter V or S) >>"), both Databases are closed, Master Sets in the Destination Database revert to their native state and this function completes with the message:

```
<><><> SAMPLE EXTRACTION COMPLETED <><><>
```

A new Source Database is then prompted for enabling a completely new sampling procedure to begin.

11.5 ERROR AND WARNING MESSAGES

ERROR OPENING DATABASE
> This message is followed by the error message returned by the DBERROR intrinsic which explains why the database could not be opened.

ERROR - Invalid Database Name. Please re-enter
> A database name cannot have more than a six character filename, and must start with a letter. The Source database name may be qualified with group and account names, which in turn must be valid.

ERROR - Extraction too complex for Classic HP and has been truncated
> Programs running on an MPE/V machine, or in Compatibility Mode on an MPE/iX machine, are limited in the data stack size available. FLEXIBASE can run out of data stack space in these programs if it has to navigate through a long chain of associated data sets to extract data (i.e. copying one detail set entry can cause a whole series of associated data entries in different data sets to be copied also). For CREATE the only solution if this occurs is to try and start the extract from a different data set, or run the program on an MPE/iX machine.

No such entry in database <<Source Database Name>>
> When transporting entries across by item value you have specified a value in a particular set which does not exist in the Source database.

<<Dataset Name>> set is full - copying incomplete
> When transporting entries across by item value or range it is possible to fill up one of the associated datasets. If this occurs the sampling will be incomplete (ie some associated entries will not have been transported). The data transported is OK, simply incomplete. If a full subset of data was required then the CREATE procedure should be done again specifying a larger capacity for the dataset, or sets, concerned. A JCW is set when this condition occurs. The JCW name is SETFULL, and it contains the value 1 if any set in the create run fills up. This allows you to detect this condition in a job stream.

IMAGE Error at line number nnnn
Dataset: <<setname>> in Database: <<database name>>
> An unexpected error has occurred while transporting data across. Should this message appear contact Open Seas with all the details (including $STDLISTs and any other relevant data).

File exists with the same name as the new root file
> There exists a file on the system with the same name as that requested for the Sample Database root file. Either choose another name, or purge this file if it is not needed.

WARNING ₵ Dataset filename conflict, cannot create database

> Although the root file can be created there exists other file(s) with names the same as one or more of the Sample Database Datasets, e.g. NEWPAY03 etc. Either remove these files or choose another name for the Sample Database.

Destination Database Name may not be qualified

> The Sample Database will be created in the logon group and account only, and so the name may not be qualified. If required, the Sample Database can be renamed upon completion using option 14 from the FLEXIBASE menu.

Entry with this key value already exists in <<sample database name>>

> When copying entries by item value ths user cannot specify a particular data item value in a specific set whose entry has already been transported across.

Data Item is not in set

> The requested Data item is not defined as belonging to the specified Dataset.

Set does not exist in database

> The requested Dataset is not defined as belonging to the specified Database.

Data item is not in database

> The requested Data Item is not defined as belonging to the specified Database.

> Schema file XPRQDASC can not successfully be put through DBSCHEMA

> With a remote source database, CREATE has to use a self-generated schema file XPRQDASC which it then uses to create the destination database via use of DBSCHEMA.PUB.SYS. Any errors in this schema file will cause this message to appear. It represents an unexpected error and should be reported to your local Open Seas office.

> If an unexpected program error occurs, such as a bounds error etc., the program will abort with the following type of message:

```
<><><><><>  Program Error Occurred in Module SAMPLEBYVALUE
<><><><><>
 STRING INDEX EXCEEDS CURRENT LENGTH (PASCERR 651)
 Program Module Trace:
     OUTERBLOCK
     SAMPLEBYVALUE
 Stack Marker:
     Index Register  %000000 (Decimal      0)
     Code Location   %004550 (Decimal   2408)
     Status          %060004 (Decimal  24580)
     Delta-Q         %000027 (Decimal     23)

To report this fault please send the following to Open Seas:
  1) Current Schema File of Database,
  2) This printout
<><><><><><><><><><><><><><><><><><><><><><><><><><><><><
><>
```

> Such an abort has not been expected, and as such represents a problem either with the software or database itself. It will require further investigation, for which purpose the items listed above should be sent to Open Seas.

[Invalid option] or [INVALID RESPONSE PLEASE RE-ENTER]

> There are some questions where there is an allowable range of acceptable answers (e.g., V or S for Value/Random Selection). If an invalid option has been entered this message is displayed and the user must re-enter one of the allowed values.

Error in << DB Call>> on Database "dname", Set Number "n"
> CREATE uses many standard IMAGE calls when operating. Any error encountered in these calls will result in this message (should never occur unless there is some problem with the database). Contact Open Seas if this error occurs.

> When specifying data value conditions a number of errors may occur. In each case CREATE will display the error and attempt to locate the error on the input line by means of a "^" symbol positioned under the offending portion. The following errors may be encountered:

[INVALID Condition - Sub Item is faulty]
> The number entered in parentheses after the data item name is invalid (perhaps greater than the maximum number of subitems.

[INVALID Condition - Relational Operator is Faulty]
> Only the relational operators "=", ">", "<", "<>", ">=" or "<=" are allowed to follow the data item name, and MUST preceed the item value.

[INVALID Condition - Item Value is Faulty]
> Perhaps a non-numeric value has been entered for a numeric item?

[INVALID Condition - Item Name is Faulty]
> The specified item name does not exist in the specified data set.

[INVALID Condition - Wildcards may only be used with "="]
> The wildcard character "@" may only be used in conjunction with the relational operator "=".

[INVALID Condition - An AND or an OR is incomplete]
> A data item name, relational operator and value has been specified followed by an AND and an OR. CREATE now expects another conditions to follow. Alternatively the user has not entered AND or OR correctly.

[INVALID Condition - A Maximum of 3 item conditions are allowed]
> Three conditions only may be entered separated by AND/OR. When entering the capacity percentage or formula the following error messages may be encountered:

[INVALID Formula - missing '*']
> When specifying a multiple of the entry count the user MUST enter ENT followed by the character "*" (indicating "multiplied by"), and then a number.

[INVALID Formula - percentage faulty]
> The percentage entered MUST be in the range 1 to 100 for capacities. For entries it must be in the range 1 to 200.

SECTION 12 ₵ UNLOAD FUNCTION

The purpose of this module is to allow the user to gain access to the data within a dataset outside of the IMAGE environment e.g. for editing with EDITOR.

The selected dataset is serially unloaded to a "flat" MPE ASCII file. The unloaded record length will equal the entry length and the number of records unloaded will equal the number of entries in the set. The unloaded record will contain the data items in the same sequence and format as in the schema definition for the set.

```
Requirements & Pre-conditions:
•   Access to the database in Mode 5 or 6 is required.
•   The user must be logged on as the database creator.
•   MPE read, write, and save access to the database group is required.
```

The following is the dialog between the user and the FLEXIBASE software:

    :RUN FLEXIBAS.PUB.HQ

Select option 11 from the menu.

```
Database name >>
```

Enter the name of the database containing the set to be unloaded. The user must be the creator of the database, and the software requires exclusive access (except when setting up a job file).

```
Set name >>
```

Enter the name of the set to be unloaded.

```
Do you want an unconditional lock applied to the
set for the duration of the unload (if not, users
may add/delete entries)? Enter Y or N >>
```

Indicate whether a lock should be applied. Bear in mind that other users would experience a lock-out for the duration of the run.

Note: The MPE data file is always built with space for 200 more entries than are in the set at the time of building. This allows some entries to be added by other users during the unloading process without the need for locking. Locking is therefore only necessary with long unloads of volatile sets or where a logically-related set of entries is required.

```
Name for MPE data file >>
```

Enter the name which you require to be given to the data file. The file must not already exist. Sufficient disc space must be available for the file to be built.

Alternatively, the unload may be made to tape or other serial device by entering "TAPE" in place of a filename.

If the user wishes to change the default of "DEV=TAPE" to another serial device e.g. cartridge tape, then a file equation should be entered preceding the run which equates the filename FXUNLOAD to the selected device e.g. :FILE FXUNLOAD;DEV=CTAPE.

```
<><><> SET HAS BEEN UNLOADED <><><>
```

This message confirms successful completion of the operation.

SECTION 13 ¢ ERASE FUNCTION

The purpose of this module is to remove all entries from a dataset. This can be a high-speed alternative to using multiple DBDELETEs. For example, it might be used before reloading a set from an MPE file created by using option 11 and to be reloaded using option 13 on the menu.

Warning ¢ this operation will not cause records to be written to an IMAGE log file even if logging is enabled for the database.

Note: erasing a detail dataset will always result in the database being logically consistent since corresponding paths in master data sets are also modified by the FLEXIBASE software. However, erasing a master set can cause the database to become logically inconsistent internally if the master set has paths to detail sets containing entries, since chain heads are removed while the detail chains remain. In practice, related detail sets MUST also be erased, whether or not the master set is to be reloaded. If the master set is to be reloaded then related detail sets should be unloaded, erased and reloaded in order to bring the database back to logical consistency.

WARNING ¢ IT IS THE RESPONSIBILITY OF THE USER TO ENSURE LOGICAL CONSISTENCY IF USING THE ERASE <u>MASTER</u> SET FEATURE. DO NOT ATTEMPT TO ERASE A MASTER SET UNLESS YOU FULLY UNDERSTAND THE ABOVE.

```
Requirements & Pre-conditions:
•   Ensure that a back-up copy of the database is available.
•   Exclusive access to the database (mode 3) is required.
•   The user must be logged on as the database creator.
•   MPE read, write, and save access to the database group is required.
•   There are no file equates referring to the database.
```

The following is the dialog between the user and the FLEXIBASE software:

```
:RUN FLEXIBAS.PUB.HQ
```

Select option 12 from the menu.

```
Database name >>
```

Enter the name of the database containing the set to be erased. The user must be the creator of the database, and the software requires exclusive access to the database (except when setting up a Job file).

```
Set name >>
```

Enter the name of the set to be erased.

```
<><><> SET HAS BEEN ERASED <><><>
```

This message confirms successful completion of the operation.

If Third Party Indexing has been enabled with DBUTIL, then FLEXIBASE will automatically erase any associated indexes.

SECTION 14 C   LOAD FUNCTION

The purpose of this module is to load entries into a dataset from a "flat" MPE file. The file loaded may, for example, have been created by using option 11 to unload a dataset C however it could also have been created by any other means.

The MPE input file may be either an ASCII or BINARY file. It must have a record length equal to the entry length of the specified set. Records from the MPE file are loaded into the dataset using normal DBPUT intrinsic calls C checks for duplicate search items, missing chain heads and full sets are therefore active (see Appendix D for how to handle exception conditions).

The set does not necessarily have to be empty before loading is commenced.

This module is compatible with IMAGE logging and therefore large numbers of log records may be written if this module is used when logging is enabled C it is suggested that logging is disabled if appropriate.

```
Requirements & Pre-conditions:
•   The user must be logged on as the database creator.
•   MPE read and write access to the database group is required.
•   The user must be able to open the database in mode 1 or 2.
```

The following is the dialog between the user and the FLEXIBASE software:

    :RUN FLEXIBAS.PUB.HQ

Select option 13 from the menu.

```
Database name >>
```

Enter the name of the database containing the set to be loaded. The user must be the creator of the database.

```
Set name >>
```

Enter the name of the set into which entries are to be loaded.

```
MPE file name >>
```

Enter the name of the disc file containing records to be loaded into the specified set. The user must have read access to the file.

Alternatively, if the data is held on a set of tape reels produced by the unload function, then enter "TAPE" in place of the filename. Other serial devices can be selected by use of a file equation as detailed in the UNLOAD function description.

```
<><><> SET HAS BEEN LOADED <><><>
```

This message confirms successful completion of the operation.

SECTION 15 ₵ RENAME FUNCTION

The purpose of this module is to rename an entire database. If the rename involves a group qualifier different from the group in which the database resides, then the database will effectively be moved into a different group (to move a database into a different account, the MPE "RESTORE ;GROUP= ;ACCOUNT= " command should be used). All files associated with the database will be renamed, i.e. the Root File, each of the datasets, the ILR file (if any) and the ANALYZE History File (if any).

```
Requirements & Pre-conditions:
•   Exclusive access to the database is required.
•   The user must be logged on as the database creator.MPE write and save access to
    the destination group is required.
```

The following is the dialog between the user and the TRANSPOSE software:

```
:HELLO
```

Log on as the creator of the database.

```
:RUN FLEXIBAS.PUB.HQ
```

Select option 14 from the menu.

```
Do you want to run Interactively now or create a streamed Job file?
      Enter I or J >>
```

The renaming can either be performed now in session mode or scheduled as a job.

```
<><><> FLEXIBASE/3000 ₵  DATA BASE RENAME <><><>
```

Identifies chosen function.

```
Old Database name >>
```

Enter the present name of the database. The user must be the creator of the database and the software requires exclusive access to the database (except when setting up a Job file). No existing database of the same name can be present, nor should any files exist with names which conflict. Both old and new database names can be qualified with a group and account name.

```
New Database name >>
```

Enter the desired name for the database.

```
WAGES.PUB.FINANCE Renamed to PAYROL.ACR.FINANCE
WAGES01.PUB.FINANCE Renamed to PAYROL01.ACR.FINANCE
WAGES02.PUB.FINANCE Renamed to PAYROL02.ACR.FINANCE
<><><> RENAME COMPLETED <><><>
```

This message confirms successful completion of the renaming. All sets are renamed, along with the ILR log file (if any) and the database history file (if any) created by the ANALYZE function.

15.1 RENAMING INDEX FILES AND SETS

MPE/iX includes a facility for Third-Party Indexing, whereby proprietary Third-Party indexing products can be added to a TurboIMAGE database. When enabled this will create indexing datasets or files (depending upon the product installed). The RENAME function will detect and rename any such index files or sets installed.

15.2 ERROR MESSAGES DURING RENAME

ERROR OPENING DATABASE
> User must be the Creator and exclusive access is required to rename.

ERROR ₵ INVALID DATABASE NAME
> The database name must be no more than 6 characters long and must start with a letter.

FILE EXISTS WITH THE SAME NAME AS THE NEW ROOT FILE
> A file or database with this name already exists.

WARNING ₵ POSSIBLE FILE NAME CONFLICT
> There are files in existence with the same names as one of the proposed new files.

<><><> RENAME TERMINATED <><><>
> The renaming process has not been completed. Any files which have already been renamed will be renamed back to their original names, where possible.

15.3 SUPPORT FOR IMAGE/SQL

Note that it is not possible to rename a data base which is ATTACHed to a DBEnvironment. Attempting to do so will result in an error message, and a detailed explanation of how you should proceed.

15.4 SUPPORT FOR B-TREE INDEX

When renaming a database which has attached B-Tree indexes the Rename module will rename the B-Tree file immediately after its associated Master set, and will display the following message on the screen:

```
/PROG/ME/SALESZ02.idx    Renamed to …….. /PROG/ME/ABC02.idx
```

In this example the B-Tree index is attached to Master Set 2, and the database is called SALESZ.ME.PROG. The file name is expressed in Posix format.

SECTION 16 C COPY FUNCTION

The purpose of this function is to perform a fast copy of an entire database. FLEXIBASE uses fast file access techniques to read an existing database and create a new copy of it, renaming it in the process. The user must have read access to the group and account where the database resides, and have write and save access to the group and account where the copy is to reside. If the source database has logging enabled then the copy database will have logging turned off and the logid removed (indeed all flags will be set to their defaults as if DBUTIL CREATE had been used). No ILR files or ANALYZE History files will be copied as these relate to the old database only, and not to the new copy database.

Fast database copies are useful for archiving and backup purposes, as well as in program development testing. The software requires the source database to be available exclusively and so this module cannot be used "on-line" while other users are accessing the database. However, once such access is available a full copy of the database can be made in another group and account, or under another name. This copy could then be used as a test database for program development, or could be used as a read-only database whilst the original is being updated so as to maximise performance. The copy could also be used as the copy for backup or archive purpose. For example, you could copy a database in job mode overnight and then store it to tape during the day (if required) thus avoiding operations staff working outside normal hours.

```
Requirements & Pre-conditions:
•    The user must be able to open the source database in mode 7.
•    The user must have read, write, and save access to the group in which the copy
     database is to reside.
```

The following is the dialog between the user and the FLEXIBASE software:

```
    :RUN FLEXIBAS.PUB.HQ
```

Select option 15 from the menu.

```
Do you want to run Interactively now or create a streamed Job file?
    Enter I or J >>
```

The actual copying of the database can either be performed in session mode or scheduled as a job.

```
Old Database name >> PAYROL.WAGES.WBB
```

Enter the name of an existing database. This name may be fully qualified, provided that the user has normal DBOPEN access to the database with regard to MPE security.

```
Enter Password >>
```

Enter a valid database password, including maintenance word, or ";" if the user is the database creator. This password must enable a DBOPEN Mode 7 call to succeed. Thus to be able to copy a database the user must know this password.

```
New Database name >>
```

Enter the database name to be used for the new copy database. This database name must not already exist, nor may any files exist with the same MPE name as any of the potential new datasets. The name may be fully qualified provided the user has write and save access to the specified group. Once accepted the software copies first the root file and then the individual sets in order, and indicates progress set by set:

```
Root File     PAYROL.WAGES.WBB    copied to NEWPAY.PUB.PROGDEV
Set Number 1  PAYROL01.WAGES.WBB  copied to NEWPAY01.PUB.PROGDEV
Set Number 2  PAYROL02.WAGES.WBB  copied to NEWPAY02.PUB.PROGDEV

<><><> DATABASE COPY COMPLETED <><><>
Further databases....(or press return to exit)....
```

```
Old Database name >>
```

Upon completion of this copy you may copy a further database, or press return to exit this function.

16.1 COPYING INDEX FILES AND SETS

MPE/iX includes a facility for Third-Party indexing, whereby proprietary Third-Party indexing products can be added to a TurboImage database. When enabled this will create indexing datasets or files (depending upon the product installed). The COPY function will detect and copy any such index files or sets installed.

16.2 ERROR MESSAGES DURING DATABASE COPYING

ERROR OPENING DATABASE
> This message is followed by the error message returned by the DBERROR intrinsic which explains why the database could not be opened.

ERROR - Invalid Database Name. Please re-enter
> A database name cannot have more than a six character filename, and must start with a letter.

File exists with the same name as the new root file
> A file or database with this name already exists.

WARNING ₵ Dataset filename conflict, cannot create database <<newdbname>>
> There are files in existence with the same names as one of the proposed new files.

<><><> COPY TERMINATED <><><>
> The copying process has not been completed. Any files which have already been copied will be purged (where possible).

16.3 SUPPORT FOR IMAGE/SQL

When the source database is ATTACHed to a DBEnvironment, then the target database will not be ATTACHed.

16.4 SUPPORT FOR B-TREE INDEX

When copying a database which has attached B-Tree indexes the Copy module will copy the B-Tree file immediately after its associated Master set, and will display the following message on the screen:

```
/PROG/ME/SALESX04.idx   copied to …….. /PROG/ME/SALESZ04.idx
```

In this example the B-Tree index is attached to Master Set 4, and the source database is called SALESX.ME.PROG. The file name is expressed in Posix format.

SECTION 17 ℂ DECOMPILE FUNCTION

The purpose of this function is to regenerate a schema file from the database itself. It may be used in situations where the original schema is lost or otherwise unavailable, or where you are not sure which of several schemas is the current version. It may also provide documentary advantage over an original schema since the format is rationalized, sets and items are numbered, and item-types and lengths are quoted within sets.

```
Requirements & Pre-conditions:
•    The user must be logged on as the database CREATOR.
•    The user must be able to open the database in shared mode
```

The following is the dialog between the user and FLEXIBASE software:

```
    :RUN FLEXIBAS.PUB.HQ
```

Select option 16 from the menu.

```
Database name >>
```

Enter the name of the database for which a schema file is required. The user must be the creator of the database, but exclusive access is not required.

```
Name for new schema disc file >>
```

Enter the name which you require to be given to the new schema file. If a file by that name already exists, then a confirmation is requested before the old file is purged. The file can be directed to other than a disc file:

Enter "ME" or "$STDLIST" or "STDLIST" to get the schema on screen in session, or on STDLIST in jobmode.

Enter "LP" to get the schema file directed to a printer, device class LP.

```
<><><> SCHEMA FILE COMPLETED <><><>
```

This message confirms successful completion of the schema.

The ASCII schema file produced is that which, if processed by DBSCHEMA, would produce the database root file for the database selected. Comments are included to give item and set numbers, and item types and lengths within each set. Any unreferenced items (items declared but not used in sets) are indicated as such in the schema, but are otherwise fully described.

NOTE: The figures in brackets after capacities are the current blocking factors. This undocumented feature of IMAGE can be used to dictate blocking factors for particular sets.

SUPPORT FOR IMAGE/SQL

It is also possible to generate scripts to handle ATTACHing the database to a DBEnvironment, and to build any additional SQL objects (such as views) that may refer to the IMAGE table descriptions. See section 24, FXUTIL Configuration Utility, for details.

SUPPORT FOR THIRD PARTY INDEXING

It is important to note that the Superdex TPI product keeps its indexes in a dataset within the IMAGE database. However, when TPI is enabled, the existence of this dataset is hidden from the user. If the

database is enabled for indexing with Superdex, DECOMPILE will therefore create a schema file which does not include the Superdex index dataset. If this schema is used as input to a run of TRANSMUTE in schema mode, it will have the effect of deleting the Superdex dataset, and thus destroying all index configurations. To produce a schema which can be used to TRANSMUTE the database without damaging the indexes, you should disable the database for indexing before running DECOMPILE. This will produce a schema in which the Superdex dataset is visible.

This problem does not affect Omnidex indexing.

SUPPORT FOR B-TREE INDEX

Decompile Schema will of course include "/Indexed" in the schema where there are B-Trees.

SECTION 18 ₵ TRANSMUTE / TRANSFER FUNCTIONS

18.1 GENERAL

INTRODUCTION

The TRANSMUTE function enables an existing database to be transformed into a new structure or format. The changes that can be made include the following:

- Changing the type or length of a data item.
- Changing the position of data items within an existing data set.
- Adding or deleting items in existing data sets.
- Changing the position of sets within a database.
- Adding or deleting data sets.
- Renaming data sets or data items.
- Adding or deleting access paths.
- Changing the capacity of a data set.
- Changing database passwords or blocking factors.

TRANSMUTE automatically identifies the need for data conversion and will retain the meanings and values of all data present in the data base during the conversion. FLEXIBASE prompts the user for information such as whether to left or right justify items that have been changed in size.

TRANSMUTE performs the database operations directly on the existing database. All changes are done in one operation and only those data sets that are affected by the changes are processed.

The TRANSFER function is like the TRANSMUTE function but in this case the data is moved from one database to another at the same time as the data is changed. The changes made are determined by the differences between the originating and receiving databases. The function can be used for creating a new version of a database while retaining the original database.

The software used in the FLEXIBASE product to perform TRANSMUTE or TRANSFER shares a lot of common code, so information in this section concerning the software can be assumed to apply to both functions unless otherwise indicated.

SCHEMA METHOD & INTERACTIVE METHOD

TRANSMUTE has two alternative methods for specifying the changes to be made:

a) The SCHEMA METHOD where all changes are specified by editing a schema file until it represents the desired final structure, then giving TRANSMUTE the name of that schema file.

b) The INTERACTIVE MENU METHOD where all changes are specified through an interactive menu-based dialog (see section 18.5).

TERMINOLOGY

The words OLD or SOURCE (as applied to the database) refer to the original database which is to be transformed or unloaded. The words NEW or DESTINATION or MODIFIED (as applied to the database or schema file) refer to the modified database or the database into which data is to be loaded, or its schema file. The word MEDIA is used to indicate a tape file, cartridge tape file, DAT tape or disc file i.e. a serial file.

THE ITEM-LIST : ITEM NAMES & SEQUENCE

TRANSMUTE requires a reference point in that all the items which exist in the old and new databases must have the same name in the old and new databases. Given this condition, TRANSMUTE can identify all data in the two databases.

SET NAMES & SEQUENCE

TRANSMUTE also requires a reference point in order to identify sets, in that all the sets which exist in the old and new databases must have the same name in the old and new databases.

NAME CHANGES

Changes to names of sets and items can be specified in the interactive menu. Generally, name changes are effected in a separate run since they are incompatible with most structural changes.

EDITING THE SCHEMA

The schema file for the database is simply edited to the required design in the normal way (e.g. using EDITOR). It is desirable at this stage to check that the schema syntax is correct, by running DBSCHEMA on the new schema file ℂ though this is also checked automatically by TRANSMUTE at the beginning of its run.

RETAINING OLD CAPACITIES (SCHEMA METHOD)

You can retain old database capacities (i.e. ignore ones in the schema) by responding "L" to the following prompt.

```
Capacities to be determined by Schema or Left unchanged? (S/L)>>
```

A reply of "L" to this question will cause NO capacities to be changed regardless of what is specified in the schema. A null capacity can also be used in the schema to protect individual sets from capacity change e.g. Capacity: ;

This capability may be useful in two circumstances. Firstly if you have used the CAPACITY CHANGE function of FLEXIBASE, then dataset capacities will get out of line with any retained schema. You can generate a correct new schema with the DECOMPILE function of FLEXIBASE although in that case any comments you had in the original schema will be lost. Using the null capacity schema capability enables you to retain the original schema without any risk of accidentally resetting capacities to the original values.

Secondly if you have multiple copies of the same database (e.g. on separate HP3000 computers) with different set capacities, FLEXIBASE can be used to set up a standard database transformation job which will make the same changes to all the databases without affecting the capacities in each database.

RETAINING OLD BLOCKING FACTORS (SCHEMA METHOD)

Blocking factor changes in detail sets require extra processing time.

```
Blocking factors to be determined by Schema or left unchanged? (S/L)>>
```

A reply of "L" to this prompt will leave ALL blocking factors unchanged (as far as is possible - some structural changes <u>do</u> necessitate the re-optimizing of some sets). This can save unnecessary processing where blocking factors may be different on different machines whilst a standard database transformation job is to be applied.

An individual blocking factor can be retained by use of a "null blocking factor" in the schema e.g. Capacity:30000();

OPTIMIZATION OF BLOCKING FACTORS (SCHEMA METHOD)

An individual blocking factor can be optimized by specifying the word "best" as the blocking factor in the schema e.g. Capacity:30000(best);

The optimization of ALL blocking factors can be invoked by setting a JCW before the run - SETJCW FXBESTBF=1.

TYPES OF TRANSFORMATION

ITEM DATA-TYPE AND LENGTH AND SUB-ITEM CHANGES

Some item changes are specified sufficiently in the schema e.g. type X8 to type J. Other item changes require further clarification to precisely specify the change e.g. type X8 to type X6 where the question: "Should truncation be from the Left or Right end of the item?" is asked by TRANSMUTE in its early stages. Sub-items can be gained or lost to/from the beginning or end of the array of sub-items C this latter point again being established by questions asked by TANSMUTE during its early stages. Refer to the section on Conversion of values to new data-types" for a specification of how ROUNDING occurs between data-types if relevant, and when OVERFLOW and UNDERFLOW can occur and their consequences. Refer to section 18.4 on "Detailed discussion of TRANSMUTE" for full details of all interactive specification of changes.

PATH & SET CHANGES

All changes are handled logically by TRANSMUTE using only the schema, but the user should bear in mind that search items in master datasets in the new database must be unique (this is a requirement of IMAGE). Changing from one search item to another in a master set could produce problems if the new search item does not have unique values. When changing or adding paths in detail sets, if a corresponding chain-head is missing from the related master set, then it will be added, whether the master set is manual or automatic. (Data other than the key in an added manual entry will be zeroized.) TRANSMUTE will warn about potential problems before beginning the physical changes. Entries containing duplicate search items (after being transformed) will be saved in MPE files for salvage purposes (see *Appendix D*.) Detail sets can be changed to master sets and vice-versa as required. New sets can be inserted in the schema C they will of course contain no entries in the new database unless they are automatic masters in which case they will be loaded normally. Sets can be left out of the new schema. New items in an entry have values of zero or spaces depending on data-type. Items can be left out of a set in the new schema.

CAPACITY CHANGES

Changes to dataset capacities can be made by simply changing the set capacity in the database schema and submitting the new schema to the transmute function. However it is recommended that the CAPACITY CHANGE function of FLEXIBASE be used to effect such changes as it is quicker and easier to use (see Section 6 of this manual).

TRANSMUTE PROCEDURE (SCHEMA METHOD)

The following is a check list of the necessary actions when using the TRANSMUTE module to restructure a database with reference to a changed schema file.

1.   Plan the changes C remember that changes to data divisions in programs may also be required, and that changes to paths or item/set numbers or names may imply changes to DB calls in programs.

2.   Edit the schema file to represent the new database.

3.   Run DBSCHEMA on the modified schema file to check the syntax.

4.   Ensure that adequate back-up is available in case it is required later by using DBSTORE, STORE or SYSDUMP to take a copy of the old database.

5. Run option 17 from the menu to convert the database and entries into the new form (this can be an interactive procedure, i.e. on a terminal, or a job file can be produced for streaming).

6. It may be advisable to run QUERY to check that the format of entries in modified datasets is as intended if item formats or lengths have been changed.

Note that TRANSMUTE does not make entries in a user log file. If user logging is enabled before a TRANSMUTE run then logging will still be enabled after the run, but structural changes will have invalidated the contents of the current log. Therefore roll-forward recovery should not be attempted after running TRANSMUTE using the current log file. You should restart the log cycle after using the TRANSMUTE function so as to create a new log file.

TRANSFER PROCEDURE

The following is a check list of the necessary actions for using the TRANSMUTE module to transfer data directly from one database into another database.

1. Plan the changes C remember that changes to data divisions in programs may also be required, and that changes to paths or set numbers or names may imply changes to DB calls in programs.

2. Run option 18 from the menu to unload entries from the old database into the other database (this can be an interactive procedure, i.e. on a terminal, or a job file can be produced for streaming).

3. It may be advisable to run QUERY to check that the format of entries in modified datasets is as intended if item formats or lengths have been changed.

Note: when using the TRANSFER option, the user MUST have write-access in the destination database to every set and item involved (failure to ensure this can yield unexpected results).

If the destination database is enabled for logging then the entries transferred in will be logged (this could result in a large number of log entries C you may prefer to disable logging before using the TRANSFER function).

CONVERSION OF VALUES

The following table shows the possible rounding, underflow and overflow that may occur when the data type of an item is changed during a TRANSMUTE run. Note that the accuracy of all conversions is limited to the accuracy of type R4 items i.e. 16 or 17 significant digits.

```
 New              Overflow     Underflow
Data-Type            Above         Below   Rounding

    J1                 9999         -9999   To whole number
    J2            999999999    -999999999   To whole number
    J4    ±9.2 X 10 POWER 18  As Overflow   To whole number
    I1                32767        -32768   To whole number
    I2           2147483647   -2147483648   To whole number
    I4      ±1 X 10 POWER 18  As Overflow   To whole number
    K1                65535             0   To whole number
    K2           4294967296             0   To whole number
    K4    1.84 X 10 POWER 19            0   To whole number
    R2                    C             C    To 5 or 6 significant digits
    R4                    C             C    To 10 or 11 significant digits
    X           Item length   Item length   To number of decimals specified
    U           Item length   Item length   As specified (0-9 decimals)
    Z           Item length   Item length   To number of decimals specified
    P           Item length   Item length   To whole number
    E2                    C             C    To 5 or 6 significant digits
    E4                    C             C    To 10 or 11 significant digits
```

When underflow or overflow occurs, the value of the item is replaced by zero and an error message is displayed to the user. For example:

```
98000 This has been zeroized as it cannot be converted to K1.
```

Rounding, underflow or overflow can also cause the uniqueness of a search item to be lost. In that case data may not be loaded (see Appendix D for recovery procedure).

CHANGING TO/FROM DATA-TYPE "E"

You can perform data-type conversions involving data-type "E" in the normal way through TRANSMUTE, which will of course convert the actual contents of the data-item involved in the entries which contain it.

If however you want to retain the data values as they are, but change the definition of the data-type in the root-file, then proceed by one of the following two methods. (This would be useful in the scenario where an application has been storing the data in IEEE format in the database, though until now the data-type could only be defined in the schema and root-file as data-type "R").

This method is fastest but requires enough spare disc space for a temporary copy of the database:

1. Use the DECOMPILE function to produce an up-to-date schema.
2. EDIT the schema to reflect the data-type change.
3. Log in to a different group.
4. Use DBSCHEMA to make a new root-file from the new schema.
5. Use DBUTIL to CREATE a new empty database.
6. STORE the root-file only to tape.
7. Use DBUTIL to PURGE the new database from the system.
8. Log in to the original group containing the real database.
9. RESTORE the root-file only from tape.

This method is slower (requires a STORE of the entire database) but requires no spare disc space:

1. Use the DECOMPILE function to produce an up-to-date schema.
2. EDIT the schema to reflect the data-type change.
3. STORE the entire database to tape.
4. Use DBUTIL to PURGE the database from the system.
5. Use DBSCHEMA to make a new root-file from the new schema.
6. Use DBUTIL to CREATE the new empty database.
7. RESTORE only the sets from tape so that they now comprise a database with the new root-file.

SHORTENING OF DATA ITEM LENGTH

Data can be lost if an item is changed to a shorter length of the same type (e.g. I4 to I2 or P20 to P12) if the new data length is insufficient to hold the full data item value. In this case the following message is reported for each occurrence of data loss:

```
Data lost in item length contraction I4 ℂ  I2
```

WORK-FILES USED BY TRANSMUTE

TRANSMUTE creates two small databases for its own use during the course of its run. These are named XPRQDA and XPRQDC. Normally these are not visible to the user since they are purged automatically during the final stage of the run, even if TRANSMUTE aborts for some reason. If however a disorderly abort occurs, then these databases may be PURGED using DBUTIL, though the next run of TRANSMUTE would do this automatically. TRANSMUTE also builds some other small work files which are all prefixed with XPRQ. Again these may be purged after a disorderly abort, if so desired, by the user of that run of TRANSMUTE.

Because these files are used exclusively, TRANSMUTE cannot be run simultaneously with itself within the same group.

RUN TIME

The run-time of TRANSMUTE or TRANSFER will vary according to the size of the database and the complexity of changes. However TRANSMUTE only processes entries from sets which are being changed (in addition, if it is solely a capacity change to a detail dataset then the dataset will simply be extended rather than processed entry by entry, unless it is a reduction of capacity to less than the previous "high-water" mark). Other unchanged sets are not processed. FLEXIBASE will also build or rebuild paths without needing to process entries. Work space is automatically claimed and purged on a set by set basis ₵ only when there is not sufficient free space on disc for the set transformation does TRANSMUTE request a tape (or tapes) to be mounted for the work files. In this way the run time and the requirement for operator intervention are reduced to the minimum possible. The TRANSFER option uses deferred output IMAGE I/O for maximum speed in updating the destination database except when ILR logging is enabled.

Because of the nature of the algorithms employed, FLEXIBASE is largely unaffected by the presence of sort items. Note that transformation of master sets is very much faster if the capacity is kept the same, and that transformation of detail sets is very much faster if the blocking factor is kept the same.

18.2 IMAGE/SQL ATTACHMENT SUPPORT

General - When an IMAGE database is ATTACHed to an ALLBASE DBEnvironment using IMAGESQL then the default attachment specification can subsequently be altered and added to, again using IMAGESQL. More specifically authorized users can be added with mode and class specifications, and items in sets can have their data representation redefined by the UPDATE and SPLIT commands. Furthermore, objects can be created in the ALLBASE DBE which are dependent on (i.e. refer to) the TurboIMAGE Tables (which are actually the SQL representation of the IMAGE sets). There are certain actions which can cause loss of these specifications. Namely an UPDATE or SPLIT command in IMAGESQL will cause the dropping and re-creating of the Tables involved in the command, and the dropping causes, in turn, all SQL objects which are dependent on those Tables to be dropped. The DETACH command in IMAGESQL causes all attachment specifications for the IMAGE database to be lost, and all the Tables (representing the sets in the IMAGE database) and their dependent objects to be dropped from the ALLBASE DBE.

When is it necessary to DETACH? It is not necessary to DETACH in order to do a capacity change or a repack, or any other FLEXIBASE action other than TRANSMUTE. However, TRANSMUTE now detects Attachment and automatically maintains that attachment, so it is never necessary for the user to DETACH for normal maintenance.

How does TRANSMUTE support attachment? – TRANSMUTE now automatically detects that the IMAGE database is ATTACHed and reports the fact at the beginning of the run.

Example:

```
Old Database name >> dpay

This Database is ATTACHED to ALLBASE/SQL.
The run will generate the necessary files and perform reATTACHment without loss
of configuration or dependent SQL objects.
```

If you then proceed to action a structural change, TRANSMUTE will first of all generate a file containing the IMAGESQL commands necessary to re-attach to all the DBEs to which the IMAGE database is currently attached, with the full current specification of users and data representation. TRANSMUTE will then generate a file containing the SQL commands necessary to recreate all the objects in the ALLBASE DBE which are user-specified and dependent on the TurboIMAGE Tables. TRANSMUTE will then automatically DETACH and proceed to do the structural changes.

```
Do you wish to proceed with the physical changes as above? - Enter Y or N >> y

Creating Logfile for Re-attachment...
Creating SQL file for Re-attachment...
DETACHING...
```

```
CLOCK-NOS        set data is unchanged.............
```

After all the structural processing has been done, both of these files will then be applied to re-attach and attempt to bring the specification of attachment and SQL objects back to its state before the run commenced. The IMAGESQL commands are applied through IMAGESQL and an output log is produced named FXATCLOG (passwords and maintenance words will be masked out). The SQL commands are applied through ISQL and an output log is produced named FXSQLLOG.

```
<><><><><><><><><><>   PHYSICAL CHANGES ARE COMPLETE   <><><><><><><><><><><>
(THU, JAN 11, 1996,  1:43 AM)

RE-ATTACHING (Examine FXATCLOG after run) ...
Re-instating dependent SQL Objects (Examine FXSQLLOG after run).....

RE-ATTACHMENT Completed
(THU, JAN  11, 1996,  1:43 AM)
:
```

After the run, the user should examine both these LOGs for problems, since a structural change may cause certain commands to be impossible to re-apply. For example if set has been deleted then any IMAGESQL commands which reference that set will fail, and any SQL objects which depend on that TurboIMAGE Table will fail to be re-created. In the case of a deletion, this is as it should be, but the inclusion or deletion of items in sets etc., can require user attention to and amendment of the Attachment specifications.

DBE Maintenance words - If the IMAGE database creator (who is the only user who can process TRANSMUTE) does not have DBA authority in the ALLBASE DBE(s) to which the IMAGE database is attached, then the user must supply DBE maintenance words to allow the process to proceed. This is done by setting an MPE environment variable, named as the fully qualified DBE name (with dots changed to underscores) to the string value of the maintenance word. In this way MPE security restrictions are upheld and the maintenance words are not exposed in any output files. For example if the database to be processed in TRANSMUTE is attached to two DBEs named "SALES.GROUPX.ACCOUNTY" with a maintenance word of "abc" and "PRODS.INFO.CORP" with a maintenance word of "xyz" then the following MPE commands should be keyed in before the TRANSMUTE run:

```
:SETVAR SALES_GROUPX_ACCOUNTY "abc"
:SETVAR PRODS_INFO_CORP "xyz"
```

If a maintenance word is omitted or incorrectly specified then the TRANSMUTE run will be unable to proceed and will terminate before any structural change with an appropriate message.

QR and QL Files - As described above, before TRANSMUTE begins a structural change, it retrieves all IMAGESQL and SQL commands as per the current attachment specification. These commands are stored in two extra permanent files with the database which remain after the run. These QR and QL files are prefixed with the database name e.g. for a database named "SALES" there will be a privileged file called SALESQR which contains the IMAGESQL commands and a non-privileged text file called SALESQL which contains the SQL commands. (The QR file has to be privileged so that the maintenance words and passwords are not visible). The QR file can be copied as a non-privileged text file by the FLEXIBASE utility program FXUTIL.PUB.HQ (menu option 3). The pair of files can also be generated at any time for a database without using TRANSMUTE, by running menu option 4 of the FLEXIBASE utility program FXUTIL.PUB.HQ. The files are provided so that the DBA has the ability to perform a full automatic re-attach in the event of an accidental DETACH. The QR file would be used as input to IMAGESQL and the QL file would be used as input to ISQL.

SPLIT and UPDATE commands in IMAGESQL - Using either of these commands on a set will cause the corresponding TurboIMAGE Table in the attached DBE to be dropped and re-created with its attendant user views. This in turn will cause the dropping of any SQL objects dependent on that Table. It is however possible using menu option 5 of the FLEXIBASE utility program FXUTIL.PUB.HQ to create an SQL file relating to a single set from a database. If this is done before the SPLIT or UPDATE is performed, then this file can be used to recreate the dropped SQL objects after the SPLIT or UPDATE, by using the file as input to ISQL.

NOTE: It is important to note that it is not possible to rename a database which is attached to a DBEnvironment.

18.3 SUPPORT FOR THIRD PARTY INDEXES

If FLEXIBASE detects that TPI indexing is enabled, then it will endeavour to modify any indexes such that they match the new structure of the database.

Changes to item or set names will be registered in the TPI product, and should reindexing be required on any dataset, then FLEXIBASE will automatically call the TPI product to carry this out.

Should an error occur for any reason, or should the TPI product be unable to service the request, then FLEXIBASE will warn you of the fact, as follows:

```
Calling TPI product to re-index set....
Error in re-indexing set - User attention required      -1
```

You should then use the indexing tool directly to resynchronize the indexes, after the FLEXIBASE run has completed.

To disable the calls to the third party product because the version is inappropriate or for some other reason, set the following JCW before the run:

```
:SETJCW FXNOTPI=1
```

Note that automatic reindexing will only be available if the database has been enabled for TPI in DBUTIL.

Note: Omnidex users should note the following limitation. Because Omnidex references items and sets by number, not by name, if any change is made to the order of items or sets during execution of the TRANSMUTE function, then this will automatically invalidate all the Omnidex indexes. If such a change is planned, then you should disable calls to TPI (see above) before running TRANSMUTE, and rebuild the indexes using the Omnidex configurator after the database has been modified.

Note: Superdex users should note the following. The Superdex TPI product keeps its indexes in a dataset within the IMAGE database. However, when TPI is enabled, the existence of this dataset is hidden from the user. If TRANSMUTE is run in schema mode using a schema file which does not include the SI dataset, then this dataset will be removed from the database, destroying the Superdex configuration (see the Section on the DECOMPILE function for more details).

If you are working in Interactive mode using the menu, then TRANSMUTE will start by exposing the SI dataset, so that it retains its correct place within the database as a whole. It is important not to alter the SI dataset in any way.

18.4 EXAMPLE OF A TRANSMUTE RUN

An example of the user dialog with the TRANSMUTE software when transforming a database (as opposed to loading data into another database) by the schema method is as follows:

```
:HELLO
```

Log on as the creator of the database to be transformed.

```
:RUN FLEXIBAS.PUB.HQ
```

Select option 17 on the menu.

```
Do you want to run Interactively now or create a
Streamed Job file?
```

```
      Enter I or J >>
```

An interactive run will take place immediately after the input stage. A job file enables the run to be scheduled by streaming.

```
<< TRANSMUTE/3000 >> Version D.00.01
(c) SECTOR 7 SOFTWARE 1986
```

Product and release level ribbon.

```
Old Database name >> SALES
```

Enter the name of the database to be transformed. Note that the user must be the creator of the database. The database name may be a fully qualified name.

```
Enter New SCHEMA Filename [or return=MENU for changes] >> SALESSC
```

Enter the name of the new schema file for the database, or just press return to cause a menu of facilities for structural change to be displayed.

```
ALL SETS & ITEMS MUST have the SAME NAMES in the
SCHEMA and in the OLD DATABASE
```

This is a reminder to the user, and is different from versions prior to D.02.01. (If you want to change names of sets and/or items, do this using the interactive menu in a separate run – name changes can not be mixed with structural changes.)

```
Capacities to be determined by Schema or Left unchanged? (S/L) >> S
```

If you want all the capacities in the database to be left unchanged, regardless of the capacities quote in the schema, then enter "L" in response to this prompt. Otherwise enter "S" and capacities will be changed as necessary to fall in line with the schema.

```
Blocking-factors to be determined by Schema or Left unchanged? (S/L) >> S
```

If you want all the blocking factors of sets in the database to be left unchanged, regardless of the blocking factors quoted in the schema, then enter "L" in response to this prompt. Otherwise enter "S" and blocking factors will be changed as necessary to fall in line with the schema.

```
Checking schema file.............(please wait)......
```

There is a pause here while the program checks the syntax of the contents of the schema file.

```
Preparing work files.............(please wait)......
```

There is a pause here while the program sets up the necessary work files.

```
Identifying ITEM changes.....
```

The program now lists new items and changes to item-types, lengths and sub-items. Where necessary further questions are asked to specify the exact nature of the change (as in this case with the number of decimal places).

```
ACCOUNT-NO change from   I2   to   R4
How many decimal places in this item?
      Enter number >> 0
```

Identifying SET changes.....

The program now lists changes to sets and gives any relevant warnings about possible problem areas in transforming as specified.

```
CUSTOMER      set change to contents
TRANSACTIONS set change to contents
TRANSACTIONS set capacity change from 23000-30000
```

Note: the disc work space required to achieve the above changes peaks at about 8000 sectors (1.8 megabytes). If this is not available when required then the program will request a tape to be mounted.

```
Do you wish to proceed with physical changes as above?
      Enter Y or N >> Y
```

After all checks are complete, the program asks whether it is required to proceed with actually transforming the database.

```
MARKETS       set is unchanged
CUSTOMER      set changing.....   1654 entries
PRODUCTS      set ℂ automatic master
TRANSACTIONS set changing.....   20169 entries
```

The program lists the sets (as per the new schema) as it processes them to build the new form of the database.

```
Purging work files.....(please wait).....

<><><> PHYSICAL CHANGES ARE COMPLETE <><><>
```

Confirmation of completion of the run.

```
:
```

18.5 EXAMPLE OF A TRANSFER RUN

An example of the user dialog with the TRANSFER software when moving (i.e loading) data into another database is as follows.

```
:HELLO
```

Log on as creator of the database to be unloaded.

```
:RUN FLEXIBAS.PUB.HQ
```

Select option 18 on the menu.

```
Do you want to run Interactively now or create a
Streamed Job file?
      Enter I or J >>
```

An interactive run will take place immediately after the input stage. A job file enables the run to be scheduled by streaming.

```
<< TRANSMUTE/3000 >> Version D.00.01
(c) SECTOR 7 SOFTWARE 1986
```

Product and release level ribbon.

```
Old Database name >> SALES
```

Enter name of database containing data to be unloaded/transferred (the user must be the creator of the database).

```
ALL SETS & ITEMS MUST have the SAME NAMES in the OLD and NEW databases
```

This is a reminder to the user.

```
New Database name >> ACCREC.ACCOUNTS.PLC
```

Enter the name of the database to receive the data (the user does NOT have to be the creator of this database).

```
Enter database Password >>
```

Enter database password (this must grant write access to the ENTIRE new database).

```
Identifying ITEM changes.....
```

The program now lists new items and changes to item-types, lengths and sub-items. Where necessary further questions are asked to specify the exact nature of the change (as in this case with the number of decimal places).

```
ACCOUNT-NO change from I2 to R4
How many decimal places in this item?
     Enter number >> 0
Identifying SET changes.....
```

The program now lists changes to sets and gives any relevant warnings about possible problem areas in transferring as specified.

```
CUSTOMER     set change to contents
TRANSACTIONS set change to contents
TRANSACTIONS set capacity change from 23000-30000
Do you wish to proceed with the physical changes as above?
     Enter Y or N >> Y
```

After all checks are complete, the program asks whether it is required to proceed with actually transferring the data.

```
MARKETS      set unloading.....   87 entries
CUSTOMER     set unloading..... 1654 entries
PRODUCTS     set is unchanged
TRANSACTIONS set unloading.....20169 entries
```

The program lists the sets (as per the receiving database) as it unloads them into the new database.

```
Purging work files.....(please wait).....
<><><> PHYSICAL CHANGES ARE COMPLETE <><><>
```

Confirmation of completion of the run.

```
:
```

18.6 DETAILED DISCUSSION OF TRANSMUTE

This section lists and discusses all possible interactions between the user and the TRANSMUTE software, in the order that the interactions typically take place. The set and item names used are, of course, purely illustrative.

```
Requirements & Pre-conditions:
•    Ensure that a back-up copy of the database is available.
•    Exclusive access to the database (mode3) is required.
•    The user must be logged on as the database creator.
```

```
•    MPE read, write, and save access to the log-on group is required.
```

```
:FILE FXUNLOAD;DEV=CTAPE
```

A valid file equation relating the media (e.g. cartridge tape) to the formal file designator FXUNLOAD may optionally be established before running the program (this is NOT however required if the TRANSFER option is to be used) else a default of TAPE is used. NO MEDIA will however need to be used in the majority of cases where the set transformations can be done on disc.

```
:FILE FXLIST;DEV=PRINTER,1,1
```

A valid file equation can be established before running the program relating the destination of the run listing (a copy of all screen interaction) to the formal file designator FXLIST, else the default LP is used.

```
:RUN FLEXIBAS.PUB.HQ
```

The program is supplied in and must stay in PUB.HQ but may be run from any account/group.

Select option 17 for TRANSMUTE or option 18 for TRANSFER from the menu.

```
Do you want to run Interactively now or create a
Streamed Job file?
     Enter I or J >>
```

An interactive run will take place immediately after the input stage. A job file enables the run to be scheduled by streaming.

```
<< TRANSMUTE/3000 >> Version D.00.01
(c) SECTOR 7 SOFTWARE 1986
```

This product title ribbon specifies the version/level number.

```
Old Database name >>
```

Enter the name of the database which is to be TRANSMUTED. Note that the user MUST be the CREATOR of the database. If using the TRANSFER option then this will be the source database name. If running interactively then this program will require exclusive access to the database. If only creation of a job file is being done then exclusive access is not required (the program will attempt to open the database in modes 5 or 6).

```
Enter New SCHEMA Filename [or return=MENU for changes] >> SALESSC
```

Enter the name of the new schema file for the database, or just press return to cause a menu of facilities for structural change to be displayed. For details of the menu operation see section 18.5.

```
File not available or does not exist
```

The stated schema file cannot be opened.

```
Unable to build workfile
```

Some usable disc space must be available to build work files.

```
Checking schema file.....(please wait).....
```

A pause occurs while the program runs DBSCHEMA on a modified copy of the schema file.

```
Syntax error in schema-file (see file XPRQLIST) or
NOROOT option used
```

There are schema syntax errors in the named schema file or the CONTROL NOROOT option is present in the schema file. Look in file XPRQLIST or use DBSCHEMA to discover the errors.

```
Unable to CREATE small working database
```

Some usable disc space must be available to CREATE a very small working database.

```
Preparing work-files.....(please wait).....
```

A pause occurs while the program creates a small working database.

```
New Database name >>
```

Enter the name of the database into which data is to be loaded. Note that this is only requested during the TRANSFER option.

```
Password >>
```

Enter the database password. This MUST allow write access to the ENTIRE new database (applies to TRANSFER option only).

```
User is not the creator of the database
```

The user must be the CREATOR of the database.

```
Database cannot be opened exclusively
ℂ  other users
```

The database must be available exclusively to this program during an actual processing run.

```
Database does not exist
```

The named database does not exist.

```
Old and New databases cannot be same for TRANSFER
option
```

The TRANSFER option can only be used on two different databases.

```
WARNING ℂ  logging is enabled on (database name)
Do you wish to proceed with logging enabled?
     Enter Y or N >>
```

Loading of data into the destination database will be logged when using the TRANSFER option if logging is enabled. This could result in a large number of log entries (use DBUTIL to disable logging if not required). This does not apply with the TRANSMUTE option.

```
Database has not been erased
Do you want to continue with run?
     Enter Y or N >>
```

There are already entries in the destination database. If the loading of data is intended to add further entries then enter Y to proceed (beware duplicate keys). Note that this only applies when using the TRANSFER option.

```
ALL SETS & ITEMS MUST have the SAME NAMES in the
SCHEMA and in the OLD DATABASE
```

This is a reminder to the user, and is different from versions prior to D.02.01. (If you want to change names of sets and/or items, do this using the interactive menu in a separate run - name changes can not be mixed with structural changes.)

```
Capacities to be determined by Schema or Left unchanged? (S/L) >> S
```

If you want all the capacities in the database to be left unchanged, regardless of the capacities quote in the schema, then enter "L" in response to this prompt. Otherwise enter "S" and capacities will be changed as necessary to fall in line with the schema.

```
Blocking-factors to be determined by Schema or Left unchanged? (S/L) >> S
```

If you want all the blocking factors of sets in the database to be left unchanged, regardless of the blocking factors quoted in the schema, then enter "L" in response to this prompt. Otherwise enter "S" and blocking factors will be changed as necessary to fall in line with the schema.

```
Identifying ITEM changes.....
```

The program now compares old and new item lists and displays details of changes and new items (referring to the new item names in the new sequence). The examples below show the dialog.

```
AGENT-CODE   new item   I2
Created item will have value of zero
```

This is a new simple item of type I2. New numeric items are initialized to zero.

```
MONTHS       new item   12 X4
Created item will be space filled
```

This is a new compound item of type X4 consisting of 12 sub-items. New character items are initialized to spaces.

```
BALANCE      item change from R2    to   R8
R8 Data-type  is not supported in value conversion
```

All data types and lengths supported in QUERY are supported here for value conversion. Above is an example of an unorthodox data length which cannot be converted. The program will not proceed with the run. Data lengths which cannot be converted are: I5 or longer, J5 or longer, K5 or longer, R1, R3, R5 or longer, P40 or longer. ALL data lengths ARE supported in the database when not changing. This only applies to TRANSFER. TRANSMUTE supports conversion between ALL data types.

```
ORDER-NO     change from  X10   to   X16
Should the extra spaces be inserted at the Left or Right of the old item?
     Enter L or R >>
```

When a character type item is to be extended, the additional spaces can be inserted at the Left end or Right end of the old item (or sub-item).

```
ADDRESS      change from  4 X40 to  4 X30
Should truncation be from the Left end or Right end of the sub-item?
     Enter L or R >>
```

When a character type sub-item is to be truncated, the characters to be lost can be taken from the Left end or Right end of the sub-item (or item).

```
ANALYSIS     change from  12  R4  to   24 R4
Should new initialized sub-items come Before existing sub-items or After?
     Enter B or A >>
```

When additional sub-items are to be incorporated into an item, this question is asked in order to establish whether the new sub-items should precede or follow the existing sub-items. New sub-items are always space filled if of data-type X or U, otherwise they are given a value of zero.

```
ANALYSIS      change from  24  R4  to  12  R4
Should sub-items be lost from the Beginning or End of the array?
      Enter B or E >>
```

When sub-items are to be lost from an item i.e. there are less sub-items in the new database item than in the old, this question is asked in order to establish whether those sub-items to be lost are the initial ones or last ones in the array of sub-items.

```
TURNOVER      change from  R4   to   X8
How many decimal places into this item?
      Enter number >>
Should preceeding zeros be zero-filled or space-filled?
      Enter Z or S >>
```

When a floating point item (E2, E4, R2 or R4) is to be converted to any other type of item, or vice versa, then a question is asked to establish the format of the new item. The new item may have from zero to 9 decimal places. Values will be rounded to the number of decimals specified. Values which are greater than the maximum possible in the new item are reported in the run and zeroized. Examples of such conversion and the other type conversions that are possible are shown below:

```
   Old Type    Old Value    User Response    New Type    New Value
                            (Dec. Places)

   R2 or R4    1234.5678          0          ZXUPIJK          1235
                                  2                         123457
                                  4                       12345678
   ZXUPIJK     12345678           0          R2 or R4  12345678.0
                                  2                      123456.78
                                  4                       1234.5678
   R2          1234.5678    No question       R4          1234.5678
   ZXUPIJK     12345678     No question       ZXUPIJK     12345678
   R4              1234          2             ZXUPIJK        123400
   R4          1234.5678          5           X8                  0
```

An overflow will be reported in the last case.

Also when a numeric item is converted to type X (character), then non-significant zeros preceeding the significant digits can be output as spaces or zeros and the user is prompted for whichever he/she wants.

```
ACCOUNT-NO change from X6  to  I
WARNING C  if conversion of any search item does
prejudice the uniqueness of the key, entries will
be saved in XPDUPE files.
```

It is permissible to change the data type of an item used as a search item. At this stage, the warning is given to alert the user, but the run continues. If the user is aware of the values to be converted and anticipates no problems with the data types used, then the warning can be ignored. If a search item value loses its unique identity because of overflow in conversion or rounding, then on loading into the new database an error will occur showing "DUPLICATE SEARCH ITEM," (see Appendix D for details) but processing will continue.

```
INDUSTRY-CODE    new item   X4
Created item will be space filled
WARNING C  this item is used as a search item. More
than one entry will prejudice the uniqueness of
the key, entries will be saved in XPDUPE files.
```

It is permissible to use an item as a search item in the new database which does not exist in the old database. However, because that item is necessarily initialized (i.e. value of spaces or zero depending upon data-type), if more than one entry is present then uniqueness is bound to be lost, and on loading into the new database, an error will occur showing "DUPLICATE SEARCH ITEM," (see Appendix D for details) but processing will continue.

```
Identifying SET changes.....
```

The program now compares set and search item structure in the old and new databases, lists all changes, and gives any necessary warnings about potential problem areas before proceeding to physically perform the changes (referring to new set names in the new sequence).

```
CUSTOMER      old set contains entries  ℂ  16054
CUSTOMER      new set capacity is       ℂ  16000
WARNING ℂ entries would be lost ℂ cannot proceed
```

Here a warning is given that data would be lost because the capacity of the relevant new set is less than the number of entries in the old set. The run cannot proceed.

```
PURCHASES     set type change D ℂ M
WARNING ℂ if entries have non-unique search-items
then entries will be saved in XPDUPE files.
```

Here a warning is given that a set is changing from a detail set to a master set. This is permitted, but if the entries do not have unique values in the chosen search item, then the message "DUPLICATE SEARCH ITEM" will be displayed ℂ see Appendix D also.

```
MARKETS       set has new search-item
WARNING ℂ if entries have non-unique search-items
then entries will be saved in XPDUPE files.
```

Here a warning is given that the set is to have a different search item. This is permitted, but if the entries do not have unique values in the chosen search item, then the message "DUPLICATE SEARCH ITEM" will be displayed ℂ see Appendix D also.

```
CUSTOMER      set capacity change from 12001-15673
```

The named set is to undergo a capacity change as stated. This has usually been explicitly requested in the schema, but in the case of detail sets small changes may arise as a result of other changes to the set resulting in optimization. Remember that capacity changes can be avoided by blanking out the capacity in the schema.

```
CUSTOMER      set blocking factor change from 6 ℂ 4
```

The named set is to undergo a blocking factor change as stated. This can arise either because the use has explicitly stated a new blocking factor in the schema, or as a result of other changes to the set.

```
CUSTOMER      set path changes
```

The named set is to undergo path changes as stated in the schema file.

```
CUSTOMER      set has changed to Auto-Expand specification
```

The named set is to have its maximum or increment changed, or Auto-Expand is to be turned ON or OFF.

```
CUSTOMER      set change to contents
```

The named set is to undergo changes to the contents of its entries, i.e. new, deleted, shuffled or changed items.

```
ORDERS        set is new ℂ will be created empty
```

The named set is in the new schema but not in the old database. It will be created as a set containing no entries.

```
TRANSACTION   set primary path change ℂ Consider
using REPACK after this run
```

The primary path in the named dataset has changed. When the TRANSMUTE run finishes, the database will be logically consistent but performance will be improved by using the REPACK function.

```
MARKETS       old set to be deleted
```

The named set exists in the old database but not in the new schema. It will not therefore be included in the new database.

Note: the disc work space required to achieve the above, peaks at about 8000 sectors (1.8 megabytes). If this is not available when required then the program will request tape(s) to be mounted.

The program states the approximate work space requirement and reminds the user that if unavailable when required then media will be needed (also if the disc space is available but unreasonably fragmented then TRANSMUTE will revert to media). It is possible to force TRANSMUTE to use media instead of disc by using the "ALTGROUP;FILES=" command to reduce the disc space available.

```
Do you wish to proceed with the physical changes as above?
     Enter Y or N >>
```

At this stage all checks are complete and the physical transformation or transfer can begin if the user so wishes.

```
CUSTOMER      set changing      1552 entries.....
```

The named set is now being transformed, the number of entries to be processed is shown.

```
CUSTOMER      set transferring  1552 entries.....
```

The named set is now being unloaded from one database to another using the TRANSFER function.

```
TRANSACTIONS set data is unchanged
```

The named set did not require any physical changes.

```
PRODUCTS      set did not exist in old database
```

The named set has been created empty since it did not exist in the old database.

```
AGENTS        set changing       357 entries.....
     (not relevant to new database ℂ ignored)
```

The named set contains no items which were in that set in the old database and therefore no entries are relevant, the set is now empty.

```
CUSTOMER      set changing   3064  ......
54000 This has been zeroized as it cannot be converted to J1
```

This particular value in an entry could not be converted to the new data type (because of overflow or underflow). The run substitutes zero for that value and continues.

```
Building path to master set 14
```

A path is now being built between the last-mentioned detail set and the stated master set. This occurs independently of any restructuring of data for the sake of performance.

```
732 new CHAIN-HEAD entries added to master set 14
```

During the building of the previously mentioned path, it was necessary to add chain-heads as stated (to either a manual or automatic master).

```
NO VACANT ENTRIES LEFT IN MASTER SET 17
```

TRANSMUTE needs to add chain-head entries to the stated set but there are no more free entries. The run has to abort.

```
DUPLICATE SEARCH ITEM, entry stored in XPDUPE03
```

A master entry cannot be put into the set because it has a duplicate search item. It is therefore stored in the stated MPE file for later salvage (see Appendix D for details).

```
File full or unavailable ₡ data has been lost
```

The duplicate entry cannot be stored for later salvage since the MPE file is full or unavailable.

```
Purging work files.....(please wait).....
```

There is a pause while the work files are purged before the program terminates. Do not interrupt this process.

```
WARNING ₡ Dataset name conflict, cannot rename database SALESA
```

The database SALESA is OK but is named XPRQDA. FLEXIBASE was unable to perform the final rename because of files with conflicting names being present (which are not part of the database) or some other similar reason. Look for such files and remove ₡ then use the FLEXIBASE RENAME function to rename XPRQDA correctly.

```
<><><> PHYSICAL CHANGES ARE COMPLETE <><><>
```

This message marks the end of TRANSMUTE. The new database is now intact as per the new schema. All changed sets are rebuilt on "DEV=DISC" and it is possible therefore that sets may have migrated from their original disc drives. If this is of concern then use the MOVE command of DBUTIL to relocate those sets.

18.7 STRUCTURAL CHANGE MENU

The menu is invoked if the user simply presses return (i.e. default) when prompted for the Schema filename. The menu gives access to all the types of change which the user may want to specify. After selecting an option from the menu, the user is generally presented with self-explanatory prompts, and LISTS from which to choose the object of the action.

For example: The user chooses option 1 on the menu - "ADD ITEM into SET".

- A LIST of SETS is displayed so that the user can choose which set.
- A LIST of All ITEMS is displayed so the user can choose the relevant item to be added into the set.
- A LIST of ITEMS in the SET is displayed so that the user can choose the position for the new item to be inserted into the SET.

Thorough checking is made to ensure that nothing inconsistent can be specified by the user - e.g. in the example above, it is not permitted to add an item into a set if that item already exists in that set.

After each option from the menu is completed, the menu is redisplayed for another choice.

Finally, instead of choosing another option, the user will enter "P" to Proceed to Processing (or "Q" to quit the specified changes).

At this point the software will proceed (in the same way as by the schema-file method) to identify and re-list the changes before asking the user for confirmation on actioning the actual physical changes to the database. If however the changes do not require physical changes to sets (i.e. simple changes such as passwords) then the changes will be applied immediately.

CHOOSING FROM LISTS

All lists consist of numbered entities.

```
E.g. 11 PRODUCT    12 ACCOUNT    13 NAME
```

The user can choose an entity by

either	- Entering the number of the entity	e.g. 12
or	- Entering the name of the entity	e.g. ACCOUNT
or	- Entering the first part of the name	e.g. A or AC

(In this case the software will pick the first entity name in the list which matches what was typed.)

This flexibility of choosing from lists has two advantages:

- The user does not have to refer elsewhere to find out what the options are - ALL options are listed.
- The user needs to do minimal typing - even 16 character names are selectable with 2 or 3 keystrokes.

PRESS RETURN FOR DEFAULT

Both for selection from lists, and other inputs, just pressing return without any input will usually cause a quitting of the option currently being input, either by going back one stage or back to the menu - whichever is more logical in the context. Where the default is different from this, the meaning of just pressing return is given in square brackets, e.g. "Select next Set whose name is to be changed [no more] >>".

NAME CHANGES TO ITEMS AND SETS

Name changes to items and sets are specified from the menu. Because name changes are in general not compatible with most structural changes, the menu will not reappear after name changes are specified. Instead the software proceeds to action the name changes immediately.

If structural changes have already been specified in a particular run, then the option for changing names will no longer be available from the menu for the rest of that run.

If creating jobfiles for both structural and name changes, then the name change jobfile must be streamed last.

SAME CHANGES TO MULTIPLE DATABASES

Using the MENU method it is simple to create a jobfile which will make the same changes to multiple copies of the database, even if the capacities and/or blocking factors are different from one copy of the database to another e.g. on separate HP3000 computers. (If the user does not specify a change to a set's capacity or blocking factor then no change will be made.)

HELP

Context sensitive help text is available at all prompts by entering "?".

DIALOG BETWEEN THE USER AND SOFTWARE

```
                       STRUCTURAL CHANGE MENU
SETS.................                    ITEMLIST..............
  1 ADD ITEM into Set                    21 CREATE NEW ITEM in Database
  2 DELETE ITEM from Set                 22 DELETE ITEM from Database
  3 Change CAPACITY                      23 Change ITEM TYPE/LENGTH
  4 CREATE a NEW SET
  5 DELETE a SET
  6 OPTIMIZE Blocking Factor            PATHS................
  7 OPTIMIZE ALL Blocking Factors        31 CREATE NEW PATH
  8 Change SET TYPE                      32 DELETE PATH
  9 REVIEW SET                           33 Change SORT-ITEM
 10 Specify AUTO-EXPAND                  34 Change PRIMARY-PATH

SECURITY..............                   B-TREE INDEXES...........
 11 AMEND PASSWORDS                      37 ASS Index
 12 ADD READ class to ITEMs             38 DROP Index
 13 REMOVE READ class from ITEMs
 14 ADD WRITE class to ITEMs            RENAME / GLOBAL ........
 15 REMOVE WRITE class from ITEMs        41 Rename DATABASE
 16 ADD READ class to SETs              42 Rename ITEMs & SETs
 17 REMOVE READ class from SETs          51 Change BLOCKMAX
 18 ADD WRITE class to SETs             52 Change NATIVE-LANGUAGE
 19 REMOVE WRITE class from SETs
                CHOOSE FUNCTION  [Q=quit  P=Process changes] >>
```

ADD ITEM into Set

```
Select the Set >>
```

Choose the Set into which an item is to be added. The set will also have its blocking-factor optimized. (OR just press return to go back to the menu).

```
Select the item to be added into the set [return=no more] >>
```

Choose an item which is to be added into this set. The item must not already exist in this set. (OR just press return to go back to the menu).

```
Insert BEFORE which item [or return=END of set] >>
```

The position of the new item in this set is determined by choosing the item before which the new item is to be placed. (OR just press return to place the new item at the end of the set).

DELETE ITEM from SET

```
Select a Set >>
```

Choose the set from which an item is to be deleted. All sets must contain at least one item, so you cannot delete the only item in a set. The set will also have its blocking factor optimized. (OR just press return to go back to the menu).

```
Select the item to be deleted from the set >>
```

Choose the item which is to be deleted from this set. You cannot delete an item which is used as a search or sort item in this set. (OR just press return to go back to choosing another set).

AMEND CAPACITY of a Set

```
Select a Set >>
```

Choose the set which is to have its capacity amended. (OR just press return to go back to the menu).

```
Present Capacity is:          20000
Current number of entries:      737  (  4% full)

Enter New Capacity >>
```

Enter the new capacity. This must be a valid positive number greater than or equal to the current number of entries. (OR just press return to go back to choosing another set).

CREATE NEW SET

```
Enter NAME for new set  >>
```

Enter the name for the new Set. The name may be a maximum of 16 characters, the first must be A-Z, the rest A-Z, 0-9 or certain other characters. (OR just press return to go back to the menu).

```
1 Manual        2 Automatic      3 Detail
Select the new set type >>
```

Choose one of the set-types for this new Set. (OR just press return to go back to the menu).

```
Do you want a B-Tree Index on this set? (Y/N) >>
```

Choosing 4 from the menu reveals a new question about whether the set should have a B-Tree Index or not, if the type of the new set is Manual.

```
Insert new set BEFORE which set  [or return=LAST SET] >>
```

The position of the new Set in the list of sets is determined by choosing the set before which the new set is to be placed. It is important to place master sets before detail sets to which they may be linked by paths. (Inserting a set into the list will change the set numbers of following sets - this might be relevant if you have applications which reference sets by number.) (OR just press return to place the new set at the end of the list .)

```
Security READ CLASSES:
Enter User-Class-No (0-63) [Return if no more] >>
```

Enter a User-class number. This must be a valid number in the range 0 to 63. You can specify any user-classes, but ultimately only those classes which have passwords assigned will be active. A user who logs-on with the creator password (;) has full read and write access to the whole database regardless. (OR just press return if you do not want to specify any further classes).

```
Security WRITE CLASSES:
Enter User-Class-No (0-63) [Return if no more] >>
```

As above

```
Select the items which are to be included in the set:
Select item to be included in set  [no more] >>
```

Repeatedly choose items to be included in the set. The same item may not be included in the set more than once. An automatic master set may only contain one item. Master sets must contain at least one simple item to be the search item. Any set must contain at least one item. (OR just press return when all items in the set have been entered .)

```
Enter Capacity >>
```

Enter the capacity. This must be a valid positive number.

```
Use "Specify AUTO-EXPAND" on the menu if expandability is required
```

This note is displayed for new detail sets (iX 5.0 onwards).

DELETE SET

```
Select the Set to be deleted >>
```

Choose from the list the set which is to be deleted. All paths into this set will also be deleted. If one of those paths was the only path into an automatic master set, then that automatic master set will also be deleted. (OR just press return to go back to the menu).

OPTIMIZE BLOCKING FACTOR of a set

ADVICE: also set BlockMax to maximum (2560) to gain full performance benefits

```
Select a Set >>
```

Choose the set which is to have its blocking-factor optimized. On MPE/iX you are advised to also increase BlockMax to its maximum to get the full benefits. On MPE/V you should follow the BlockMax recommendation (if any) from the DIAGNOSE module in FLEXIBASE. (OR just press return to go back to the menu .)

```
Do you want to optimize the blocking factor (Y/N) >>
```

If you want to confirm that the set is to have its blocking factor optimized then enter "Y", otherwise enter "N" and the set will not be optimized.

OPTIMIZE ALL BLOCKING FACTORS

ADVICE: also set BlockMax to maximum (2560) to gain full performance benefits

```
Do you want all blocking factors optimized? (Y/N) >>
```

When Optimizing All blocking-factors, on MPE/iX you are advised to also increase BlockMax to its maximum to get the full benefits. On MPE/V you should follow the BlockMax recommendation (if any) from the DIAGNOSE module in FLEXIBASE. Because many sets may need to be rewritten and paths rebuilt, you should consider the available run time for this processing. If you want to confirm that All sets are to have blocking factors optimized then enter "Y", otherwise enter "N" and the sets will not be optimized.

CHANGE SET-TYPE

```
Select the set to have its type changed >>
```

Choose the Set of which you want to change the type. You can only change to or from a detail set if there are no paths in the set. You can only change to an automatic master if the set contains one item only and at least one path. If changing from or to a detail, then the blocking-factor will be optimized. (OR just press return to go back to the menu.)

```
Currently this set is of type DETAIL

  1 Manual        2 Automatic     3 Detail

Select the new set type >>
```

Choose the new type for this set. You can only change to or from a detail set if there are no paths in the set. You can only change to an automatic master if the set contains one item only and at least one path. If changing from or to a detail, then the blocking-factor will be optimized. (OR just press return to go back to the menu.)

```
Select which item is to be the SEARCH-ITEM >>
```

Choose which item is to be the search-item of the new master set.

```
Use "Specify AUTO-EXPAND" on the menu if expandibility is required
```

This note is displayed for sets made into detail sets (iX 5.0 onwards).

REVIEW SET

```
Select set to be reviewed >>
```

Choose which set to review. The specification of the set will be displayed in schema-format with item-types and the current number of entries commented in. The blocking factor is displayed in brackets after the capacity – if "best" then it will be optimized in processing, if empty then it will not. (OR just press return to go back to the menu.)

```
Name:DEPARTMENTS      ,Manual(5/10);
Entry:
  DEPT          ( 1),                            <<    K  1 >>
  PAY-TYPE        ,                              <<    X  2 >>
  DEPT-NAME       ,                              <<    X 24 >>
  MANAGER         ,                              <<    X 24 >>
  COSTED-GROSS    ,                              <<    R  4 >>
  COSTED-ER-NI    ,                              <<    R  4 >>
  COSTED-EE-PENS  ,                              <<    R  4 >>
  HOLS            ;                              <<    R  2 >>
Capacity:          321 (best) ;  <<       67 Entries >>


(Press Enter to return to menu)
```

```
Capacity: 10000 (25), 3000, 1000; << 2467 >>
```

Sets which are specified for Auto-Expand have the capacity line formatted as above, with schema conventions. The figures are, from left to right, maximum capacity, (blocking factor), current capacity, increment; <<current entries>>.

SPECIFY AUTO-EXPAND FOR DETAIL SETS

```
Available SETS in database are:
1  EMPLOYEES     2  LOCN/DEPTS     3  BACS-DATA     4  **All Sets**

Select a Set  >>
```

Choose the Detail set which is to have its Auto-Expansion specified or changed. You can also choose "**All Sets**" and apply the specification to all detail sets in the database. Master sets cannot be made Auto-Expandable. (OR just press return to go back to the menu.)

```
Maximum Entries:    600000
Increment Entries   50000

Enter Maximum Capacity (0=disable)  >>
```

The current specification (Maximum and Increment), if any, is first displayed. Enter the maximum number of entries up to which auto-expansion is to be allowed. The absolute maximum is 16.77 million multiplied by the blocking factor. Alternatively, entering zero will disable auto-expansion. (OR just press return to go back to select a set.)

Note that specifying a reduction of an existing maximum is acceptable and will work correctly, though the reduction will not be observable in a LISTF.

```
Enter Increment as Entries or Percent (default = 10%)  >>
```

Enter the number of entries, or the percentage of the current capacity, by which the set is to be expanded each time it becomes full. A percentage must be terminated by the percent sign. Thus for example enter 250000 or 15%, or just press return for the default of 10%. A percentage is in fact converted to a number of entries by FLEXIBASE and stored in the database as such.

AMEND/DELETE/ADD PASSWORDS

(REMEMBER - Capitals and lower case are relevant in passwords.)

```
Current User-Classes and Passwords are:
 5   WALL         10   PLUM

Enter Class-no to amend/delete, or New class-no (0-63) >>
```

Enter the User-class number of the password which you want to AMEND or DELETE, OR enter a NEW user-class number to which you will next assign a password. The user-class must be a valid number in the range 0 to 63. (OR just press return to go back to the menu.)

```
Enter New Password ( or just press return to Delete) >>
```

Enter a new password for the selected user-class number. Capitals and lower case ARE relevant in database passwords, any characters may be included. Passwords may be from 1 to 8 characters in length. (OR just press return to delete the user-class from the password list.)

ADD/REMOVE READ/WRITE class to ITEM/SETs

```
Current User-Classes and Passwords are:
 5   WALL         10   PLUM

Select one or all of the listed items >>
```

Choose the Item (or set) on which you want to amend the security-access. OR choose the listed option to amend security-access on ALL items (or sets). (OR just press return to go back to the menu.)

```
Current Security is:   READ User-Classes / WRITE User-Classes
                                   5/10
Enter User-Class-No (0-63)  [Return if no more]  >>
```

Enter a User-class number. This must be a valid number in the range 0 to 63. You can specify any user-classes, but ultimately only those classes which have passwords assigned will be active. A user who logs-on with the creator password (;) has full read and write access to the whole database regardless. (OR just press return if you do not want to specify any further classes.)

CREATE NEW ITEM in database

```
Enter NAME for new item  [return=no more] >>
```

Enter the name for the new Item. The name may be a maximum of 16 characters, the first must be A-Z, the rest A-Z, 0 9 or certain other characters. (OR just press return to go back to the menu.)

```
Enter number of SUBITEMS if compound [return=simple item] >>
```

Enter the number of sub-items if the item is to be compound. (A search-item may not be compound.) (If amending the number of sub-items then dialogue later in the run will define how existing data is to be distributed.) Total item length (length*sub-items) may not exceed 2048 words. The number of sub-items may not be 0 or exceed 255. (OR just press return if the item is to be simple (=1 subitem).)

```
Enter Item DATA-TYPE >>
```

Enter the item-type. This may be one of the characters X,U,I,J,K,R,P,Z or E. Do not use type E (IEEE Real) on releases prior to MPE/iX4.0. If the item is to be a sort-item it may only be of types X,U or K. (OR just press return to go back to choosing an item for type-change.)

```
Enter item LENGTH >>
```

Enter the item (or sub-item) length. For type P this is nibbles and must be a multiple of 4. For types X,U,Z this is bytes and must be a multiple of 2. For other types this is 16-bit halfwords. Total item length (length*sub-items) may not exceed 2048 halfwords. (If amending existing length then dialogue later will establish truncation etc) (OR just press return to go back to the menu.)

```
Insert BEFORE which item  [or return=END of itemlist] >>
```

The position of the new Item in the Item-list is determined by choosing the item before which the new item is to be placed. (Inserting an item into the list will change the item numbers of following items - this might be relevant if you have applications which reference items by number.) (OR just press return to place the new item at the end of the list.)

```
Security READ CLASSES:
Enter User-Class-No (0-63) [Return if no more] >>
```

Enter a User-class number. This must be a valid number in the range 0 to 63. You can specify any user-classes, but ultimately only those classes which have passwords assigned will be active. A user who logs-on with the creator password (;) has full read and write access to the whole database regardless. (OR just press return if you do not want to specify any further classes.)

```
Security WRITE CLASSES:
Enter User-Class-No (0-63) [Return if no more] >>
```

As above

DELETE ITEM from database

```
Select an item to be deleted  >>
```

Choose the item which is to be deleted from the database. The item must not currently be included in any set i.e. it must be unreferenced. If the item is still referenced then use the "Delete Item from Set" menu option first. (OR just press return to go back to the menu.)

CHANGE ITEM TYPE/LENGTH

```
Select Item to be changed >>
```

Choose the item to have its data-type changed. Sets containing the item will have their blocking-factors optimized. (OR just press return to go back to the menu.)

```
Current (subitems,) type and length are:    R  2

Enter number of SUBITEMS if compound [return=simple item] >>
```

Enter the number of sub-items if the item is to be compound. (A search-item may not be compound.) (If amending the number of sub-items then dialogue later in the run will define how existing data is to be distributed.) Total item length (length*sub-items) may not exceed 2048 words. The number of sub-items may not be 0 or exceed 255. (OR just press return if the item is to be simple (=1 subitem).)

```
Enter Item DATA-TYPE >>
```

Enter the item-type. This may be one of the characters X,U,I,J,K,R,P,Z or E. Do not use type E (IEEE Real) on releases prior to MPE/iX4.0. If the item is a sort-item it may only be of types X,U or K. (OR just press return to go back to choosing an item for type-change.)

```
Enter item LENGTH >>
```

Enter the item (or sub-item) length. For type P this is nibbles and must be a multiple of 4. For types X,U,Z this is bytes and must be a multiple of 2. For other types this is 16-bit halfwords. Total item length (length*sub-items) may not exceed 2048 halfwords. (If amending existing length then dialogue later will establish truncation etc) (OR just press return to go back to the menu.)

CREATE NEW PATH

```
Available SETS in database are:
1 CLOCK-NOS  2 DEPARTMENTS  3 DUMMY  4 LOCATIONS
Select Master Set for path >>
```

Choose the Manual or Automatic master set which is to contain the chain-heads. The set will also have its blocking factor re-optimized. (OR just press return to go back to the menu.)

```
Available SETS in database are:
1 EMPLOYEES  2 LOCN/DEPTS  3 BACS-DATA

Select detail set for path >>
```

Choose the Detail Set within which the entries will be linked to the chosen master set. The set will also have its blocking factor re-optimized. (OR just press return to go back to the menu.)

```
Possible SEARCH-ITEMS in this set are:
1 UNION  2 GRADE-CODE

Select item to be the search-item in the detail set >>
```

Choose the Item in the detail set which is to be the search-item for this path. Only items in the detail set having the same data-type as the master search-item are listed. (OR just press return to go back to the menu.)

```
Possible SORT-ITEMS in this set are:
1 CLOCK-NO  2 DEPT  3 STATUS  4 NAME  5 NI-NO
Select item to be the sort-item (if required) >>
```

Choose the Item in the detail set to be the sort-item for this path (if a sort-item is required). Only items in the detail set with suitable data-types (i.e. X,U or K) are listed. (OR just press return if NO sort-item is required on this path.)

DELETE PATH from Database

```
PATHS in this database are:
  Pr Master Set      Detail Set      Search Item     Sort Item
  1   CLOCK-NOS       EMPLOYEES       CLOCK-NO
  2 ! DEPARTMENTS     EMPLOYEES       DEPT            CLOCK-NO
  3   CLOCK-NOS       EMP-DATA        CLOCK-NO
Select one of the listed paths >>
```

Choose from the list the path which is to be deleted. If the path is the only path into an automatic master set, then that automatic master will also be deleted from the database. Sets referenced by the path will have their blocking-factors optimized. (OR just press return to go back to the menu.)

CHANGE SORT-ITEM

```
PATHS in this database are:
  Pr Master Set      Detail Set      Search Item     Sort Item
  1   CLOCK-NOS       EMPLOYEES       CLOCK-NO
```

```
    2 ! DEPARTMENTS      EMPLOYEES      DEPT           CLOCK-NO
Select the path on which the sort-item is to be changed >>
```

Choose the path on which the sort-item is to changed, deleted or added. (OR just press return to go back to the menu.)

```
Possible SORT-ITEMS in this set are:
1 DEPT  2 STATUS  3 NAME  4 NI-NO  5 NI-CLASS

Select the new sort-item [or return to delete sort item] >>
```

Choose which item is to be the new sort-item. The list will only contain the items in the set of a suitable type - X,U or K. (OR just press return if the path is to have no sort-item.)

CHANGE PRIMARY-PATH

```
Available SETS in database are:
1 NEWNAME  2 EMPLOYEES  3 BACS-DATA  4 EMP-DATA

Select the set in which the primary-path is to be changed >>
```

Choose the Detail Set in which the primary path is to be changed. The primary path is the path on which REPACKING can take place - see section on REPACK Function. (OR just press return to go back to the menu.)

```
PATHS in this set are:
  Pr Master Set       Detail Set      Search Item     Sort Item
  1   CLOCK-NOS        EMPLOYEES       CLOCK-NO        STATUS
  2 ! DEPARTMENTS      EMPLOYEES       DEPT            CLOCK-NO

Select the new primary path >>
```

The "!" indicates the current primary path. Choose the path which is to be the primary path. The primary path is the path on which REPACKING can take place - see section on REPACK Function. (OR just press return to go back to the menu.)

ADD B-TREE INDEX

```
Available SETS in database are:
1 NEWSET            2 CLOCK-NOS        3 DEPARTMENTS       4 DUMMY
5 LOCATIONS         6 EMP-DATA

Select the Master Set to have B-Tree Index >>
```

Chosen sets are validated to not already have a B-Tree Index. Only Master (Manual or Automatic) are able to have B-Trees (not Detail Sets).

DROP B-TREE INDEX

```
Available SETS in database are:
1 NEWSET            2 CLOCK-NOS        3 DEPARTMENTS       4 DUMMY
5 LOCATIONS         6 EMP-DATA

Select the Master Set to have B-Tree Index >>
```

Chosen sets are validated to have a B-Tree Index.

SCHEMA

B-Trees can be specified in a schema using the standard syntax (/Indexed).

PROCESSING

The creation, dropping and rebuilding of B-Trees is all handles transparently, robustly and in the most efficient way (e.g. B-Trees not rebuilt for sets which are being modified unless the search-items is implicated in the changes or the capacity is increasing). A few new progress messages may be observed.

```
Dropping Appropriate B-Trees

Building modified or New B-Trees

Re-establishing B-Tree Settings
```

The latter message refers to the settings of BTREEMODE1 and WILDCARD in DBUTIL, which are re-installed at the end of the TRANSMUTE run.

RENAME DATABASE

```
Current Name of database is PAYROL

Enter New Name for database >>
```

Enter the new name for the database. The name may be up to 6 characters in length, the first must be A-Z, the rest A-Z or 0-9. (OR just press return to go back to the menu.)

CHANGE BLOCKMAX

```
On MPE/XL and MPE/iX Set BlockMax to maximum (2560) to get best performance

Current Value of BlockMax is 2560

Enter new value for BlockMax (Range is 128 TO 2560 words) >>
```

Enter the new value for BlockMax in words. The range is from 128 to 2560, though that range may be less on earlier versions of MPE. For MPE/iX you are advised to use the maximum. For MPE/V you are advised to use the value recommended in the DIAGNOSE module of FLEXIBASE, since too high or too low values will impair performance. There is no point changing BlockMax unless you also Optimize blocking-factors. (If you reduce the blockmax from its current value there is a high probability that current blocking factors will become unworkable and this run will fail – therefore if you reduce BlockMax then also Optimize All blocking-factors.) (OR just press return to go back to the menu.)

CHANGE NATIVE-LANGUAGE

```
Current Language is NATIVE-3000
1 NATIVE-3000  2 AMERICAN  3 CANADIAN-FRENCH  4 DANISH  etc
Select the new language >>
```

Choose the new language to be native for this database. Sorted chains will not be rebuilt to match the new native sorting sequence. If you want the sorted chains rebuilt, remove the implicated path in this run, then reinstate the path in another run. (OR just press return to go back to the menu.)

SECTION 19 C ARCHIVE FUNCTION

19.1 PURPOSE AND GENERAL PRINCIPLES

The purpose of the ARCHIVE function is to extract subsets of entries in a database into a smaller, archive version of the database. It selects a limited number of records from a source database, puts them into an archive database and deletes them from the original database. The new database will be a logical subset of the original database. The archive database is created in the log-on group during the run, if it does not already exist (see below). The user must have the appropriate access to the group and account of the main source database to be able to DBOPEN the database for read and write (i.e. DBGET and DBDELETE) access. The archive database will have all flags (logging etc.) set to their defaults. Remote databases can also be archived from, but in this case direct access to the source root file is not possible and so passwords and read/write lists are set to default values.

The primary purpose of the ARCHIVE module is to archive logically consistent data into new archive databases. By enabling archiving of all logically connected entries a consistent subset of data entries will be created and will be deleted from the source database. For example, if an accounts database contains data for more than one company it will be possible to archive those entries relating to one of the companies into another, smaller, database.

Before configuring an ARCHIVE run, it is necessary to grasp three notions which are fundamental to the archiving process:

- The "key" dataset. This is the initial dataset from which entries will be extracted for archiving. For each entry selected in the key dataset, the ARCHIVE function will also extract entries in other datasets that are associated with it by paths. For example, if the key dataset is a detail, then for each entry selected, any entries in the associated master dataset(s) will also be extracted for archiving. For each archiving run, you must designate a single key dataset.
- The archiving subset. By default, the ARCHIVE function will follow paths through the entire database, to extract entries from the key set, and from all its associated sets. However, it may not be required to archive data from all the associated sets. It is therefore possible to define an "archive subset". This is done simply by designating the master sets at which the archiving function should stop when following paths through the database (you must select master datasets here, since to archive an entry from a detail dataset, you must necessarily archive its corresponding chain headers also).
- "Non-delete" datasets. It may be required to archive entries from some datasets, without deleting them from the source. A typical example might be a detail dataset of customer orders, associated with a master dataset of customer details. In this case, we would want to archive and delete entries in the orders dataset, along with their master chain header entries; however, we would not want to delete these chain headers from the customer dataset. It is therefore possible to declare one or several "non-delete" sets within the archive subset.

```
Requirements & Pre-conditions:
•   The user must be able to open the source database in mode 3.
•   The user must supply a password which gives complete read and write access to
    ALL sets and items in the source database.
•   The user must have read, write, and save access to the log-on group, or to the
    group in which the destination database resides.
```

19.2 USER DIALOG

The following is the dialog between the user and the FLEXIBASE software:

```
:RUN FLEXIBAS.PUB.HQ
```

Select option 19 from the menu.

```
Do you want to run Interactively now or create a streamed Job file?
     Enter I or J >>
```

The actual creation of the archive database can either be performed in session mode or defined and run as a job.

```
Source Database Name >>
```

Enter the name of any existing database from which you wish to archive entries into the Archive Database. This Database Name may be fully qualified and you must supply a password which will give full access to all parts of the database from which you wish to archive. File equations and remote databases are allowed. Note that ARCHIVE will fully qualify the source database name before attempting to open it, thus if a file equation is required it must relate to the fully qualified database name. For example, if the program is run while logged into the PUB group of the HQ account and the source database name TESTDB is entered then ARCHIVE will open the database TESTDB.PUB.HQ. If the user back-references this to another database he must enter the file equation :FILE TESTDB.PUB.HQ=<<other database name>>. Note: As long as the file name on the left hand side of the file equation is fully qualified, it can be any valid MPE file name (e.g. X.Y.Z or DB.GROUP.ACCT).

```
Password >>
```

Enter password which gives full read and write access to all parts of the source database. If the source database is a remote database then this password will be the only known password from the source database, all others will be defaulted. Thus, this password <u>must</u> give full read access to the entire source database. In any event the password should give read access to all of the source database to enable complete copying of entries, to enable the correct set capacities to be used and to avoid inexact copying of the source database.

```
Archive Database Name >>
```

Enter a valid database name for the new Destination Archive Database. This will be created as an exact copy of the Source Database (unless remote) but with reduced Capacities (except in the case of a remote source database – see above), and will be created in the logon group. The database name may not be qualified and must not already exist.

```
Enter Capacity Percentage or Formula >>
```

Enter the percentage of the Source Database Set Capacities to be used in the Destination Archive Database. You may also express this as a multiple of the number of entries in the Source Database. Thus entering the number "50" will cause the new database to be built with half (50%) of the capacities of the Source Database. Specifying a number on its own instructs ARCHIVE to calculate the Archive Database's Capacities as a multiple of the Source Database's Capacities. Thus, entering 10 will cause the Destination Capacities to be one tenth (10%) of the Source Capacities. The value entered must be greater than zero and less than or equal to 100. To use the Source Database's entry counts as the basis for the calculation enter "ENT*" followed by the percentage required. Entering "ENT*50" will specify 50% of the entry counts of the Source Database. Thus, to base the percentage upon the Capacities simply enter a number between 0 and 100. To base the percentage upon the number of entries enter "ENT*" followed by a number between 0 and 200. In this way the Archive Database may be created with Capacities up to twice that of the number of entries in the Source Database.

Having specified the overall capacity percentage, the newly-calculated capacities of the sample database are now displayed. Individual set's capacities may be altered as required. Thus you may override the automatic calculations if required.

```
The Archive Database will have the following Capacities:

     Dataset Name   Type   Capacity
     CLOCK-NOS       A       157
     DEPARTMENTS     M        23
     BANKS           M        37
     DUMMY           M         1
     EMP-DATA        M       157
```

```
Do you want to amend any of these Capacities (Enter Y or N) >>
```

Enter Y to amend individual sets' capacities by specifying the required capacities.

```
Dataset Name to amend >>
```

Enter the dataset name whose capacity you wish to modify from the value displayed above.

```
Capacity Requested >>
```

Enter the requested capacity for this set, e.g. 127.

```
Creating Archive Database.....(please wait).....
```

If in session mode then the new archive database will be created at this point. In job mode the database is not created and any future validation is done against the Source Database. If the Source database is located on a remote machine then at this point the following lines are displayed on $STDLIST:

```
N.B As SOURCE DATABASE is REMOTE, default passwords of 5 READER & 10 <<pass>>
--- and default read/write lists (5/10) will be inserted in the archive database.
```

In the case of a remote source database, passwords are not available and thus the item and set read/write lists are set to defaults of (5/10). Password level 5 is defaulted to the password READER, and password level 10 is set to the password used to open the source database previously. If the remote source database was opened with the ";" password then a password 10 WRITER is used.

In order for ARCHIVE to navigate through the database structure when copying entries it also temporarily changes all Automatic Master Sets in the destination database into Manual Master Sets. At the end of the run they are reset back to Automatics. The user should NOT, however, break and abort an ARCHIVE run as the Automatic sets will not have been changed back to Manuals. This is very important.

```
Are Archived Entries to be DELETED from the Source database? (Y/N) >>
```

ARCHIVE will delete entries from the Source database. Since the result of this operation may be difficult to predict before the run, the user may choose not to delete entries from the Source database, thus enabling the user to "try out" the ARCHIVE run initially. The resulting archive database will contain all entries selected from the Source database, but none of these entries will have been deleted from the Source database, leaving it intact.

The next three questions are only asked if the answer to the above question is Y.

```
Should data be RECOVERED in the event of a run-time error? (Y/N) >>
```

ARCHIVE will delete Detail Set entries as it successfully archives from the Source database (and master entries once selection is complete). If an error occurs in archiving, e.g. a set in the archive database fills up, ARCHIVE can replace all those deleted Detail set entries back in the Source database – thus leaving it intact. Answer Y if you want to enable this data recovery. Answering N will leave both Source and Archive databases incomplete.

```
Should Image LOGGING be turned off in the Source database? (Y/N) >>
```

Because ARCHIVE deletes entries from the Source database using normal IMAGE intrinsics these deletes will be logged to any IMAGE Log File that is in use. This may result in a very large log file and you may select ARCHIVE to disable logging for the duration of the run and re-enable logging at the end. Note that this will result in an inconsistent log file so it is recommended that a new logfile should be built after the archiving run is complete. The next two questions allow the user to define "non-delete" datasets as described in "Purpose and General Principles" above.

```
Do you wish to DELETE only from selected sets? (Y/N) >>
```

Enter N and ARCHIVE will delete ALL entries from ALL datasets that it archives from the Source database. This may not be acceptable, the user requiring only certain sets to be deleted from (e.g. only one Detail Set, but none of its associated Master Sets). Enter Y to select a list of all of the dataset names from which entries ARE TO BE DELETED from the Source database.

You may now select a list of dataset names (Masters or Details) from which entries ARE TO BE DELETED from the Source database. Terminate the list with a null entry. Note that master set entries can only be deleted if all of their associated Detail entry chains on all its paths are also deleted. As many dataset names as required may be entered into this list, the dataset must exist in the Source database.

The dialog is as follows:

```
Set Name to delete entries from [press return to end] >>
```

You may now select entries from the Source Database to be archived into the Archive Database. Specify a "key" dataset (as described in "Purpose and General Principles" above) and particular values of any item in that set and the matching entries will be archived, along with as many associated detail/master entries as possible. Up to three data item value conditions may be entered, and the wildcard character "@" may be used to refine those entries to be archived. It is important to understand the process which ARCHIVE uses to archive entries across. Starting with the "key" dataset specified, ARCHIVE will identify the entry or entries in that set matching the data-item value conditions entered. If this set is a detail set then ARCHIVE will also have to archive entries in those master sets which have paths into the detail set (to provide the correct chain heads). These master sets may have other detail sets associated with them by paths – the entries in these detail sets for the relevant values will also be archived. These details may themselves require master set entries to be archived to provide chain heads, and so the whole process repeats itself until all paths are exhausted, or sets fill up.

Provided the Destination Set capacities are large enough, ALL logically connected data set entries will be archived. There is no practical limit to the number of sets involved, nor the number of entries archived.

```
Dataset Name >>
```

Enter the name of any of the Datasets in the Source Database from which you want to archive entries with matching data-item values. Any Manual Master, Automatic Master or Detail Sets associated to this set by paths will have their associated chains or chain heads archived also (along with their associated entries). Any set may be specified, even Automatic Masters (as these are temporarily modifed to become Manual Masters for the purposes of the ARCHIVE run. The set should be selected carefully so as to ensure as complete archiving as possible.

```
Enter Data Value Condition(s) >>
```

The user may now specify up to three data item value conditions which will be used to select matching data set entries. Each condition is composed of a data item name, a relational operator and a value. If more than one condition is entered then they must be separated by an "AND" or an "OR".

Thus, the general syntax can be represented as follows:

```
ITEM RELOP VALUE
[AND/OR ITEM RELOP VALUE
[AND/OR ITEM RELOP VALUE]]
```

where:

ITEM represents the name of any of the Data-items in the specified dataset, and may include the specification of a specific sub item number (enclosed by parentheses).

RELOP is one of the allowed relational operators ">", "<", "=", "<>", ">=" or "<=". Spaces may be entered around relational operators, but are not mandatory and are ignored.

VALUE is a relevant value for the specified data item. The wildcard character "@" may be used to specify ranges of values.

AND/OR is either the words AND or OR. These are mandatory if more than one condition is entered.

For example, to select all entries which have a data value of 20000 for the ACCOUNT-NO item in the specified set (assuming ACCOUNT-NO to be a valid item within the specified set) the user would enter:

```
ACCOUNT-NO=20000
```

This may also be entered with spaces, eg ACCOUNT-NO = 20000; or with quotes, eg ACCOUNT-NO="20000". The spaces and quotes are ignored, except in the instance when the quotes may be used to specify preceding spaces in a data item value. Thus, REFERENCE=" A" is different to REFERENCE="A".

Selecting on multiple (up to three) data value conditions is a powerful way of refining the specific entries required. Each condition must be separated by AND or OR, further expanding the possibilities.

Thus, to sample those entries with a market-code of IT and an industry-code of BRICK the user would enter:

```
MARKET=IT AND INDUSTRY=BRICK
```

Note that the data item names, and AND/OR, may be in upper or lower case (or even a mixture of the two) but the data item values must be in the appropriate case to be matched correctly.

The following are valid examples of data value conditions (assuming that the items specified exist in the specified set:

```
NAME > "A" and NAME < "B"
AMOUNT >= 0 AND amount <= 200
COMPONENT(12)<>" "
MARKET=IT or Market=UK or market="FR"
INDUSTRY<>"SAN" AND INDUSTRY>"REFR"
ADDRESS(2)>XXXXXX
```

The wildcard character "@" may be used to further refine the selection criteria. Thus entering STOCK NO="PAP@" will select only entries with a STOCK-NO whose first three characters are PAP. Each of the three potential conditions may include the wildcard character - but it may only be associated with the "=" relational operator. Thus SUPPLIER > A@ is invalid as it does not use the "=" operator.

Examples of its valid use are:

```
SUPPLIER=COMP@
ACCOUNT-NO = 2@ AND MARKET <> UK
BALANCE>0 or NAME=ICI@
DATE=@91
```

The next two questions allow the user to define an archive subset, as defined in "Purpose and General Principles" above.

```
Restrict this Archive to a subset of sets? (Y/N) >>
```

Enter N and ARCHIVE will select ALL qualifying entries from the selected set as well as all linked entries from all associated datasets. This may be a long process, difficult to predict, and resulting in a large quantity of entries some of which may not be relevant. Detail dataset entries must also archive their associated Master entries to be archived successfully, but archiving of further Detail chains can be prevented by specifying these Master sets as terminating datasets. For example, if archiving entries from a detail set with only one master path (which in turn has four other dependant detail sets) the user can restrict

the archive to only the detail and its master set by identifying the master set as a terminating set. None of the other dependant detail sets will be archived. Enter Y to specify a list of Master sets at which to terminate this process.

```
Terminating Master Set Name [press return to end] >>
```

You may now select a list of Master dataset names at which to stop the recursive extraction process by paths. Detail entries must archive their associated Master entries to be archived successfully, but the archiving of further Detail chains can be prevented by specifying these Master sets as terminating datasets. As many set names as required may be entered, the set name must exist in the Source database and must be either a Manual or Automatic Master set. Terminate this list of set names with a null entry.

The software now attempts to archive entries from this set and any sets connected by paths to this set (and any sets connected to them by paths, and so on). For example, suppose the requested set is a Manual Master and one condition is entered (e.g. Search-Item=123456). This Master set has two Detail Sets, each of which has one other Master Set (which are not connected to any other sets). The result would be: one entry archived from the Manual Master Set (assuming the entry exists with the requested value); both Detail Set chains for this Search Item value (if any, and as long as the Detail Sets do not become full); any entries in the other two Master Sets required to provide chain heads for the Detail Set chains.

The output would look like the following:

```
Archiving entries by Range.....
             Num Entries Num Entries Percentage Percentage
Dataset Name  Archived    Deleted    Deleted    Set Full

MAN1              6           6        100%        26%
MAN2              3           3        100%        60%
MAN3              3           3        100%        60%
```

In this example, the "Percentage deleted" may be less than 100% in cases where a partial archive is requested, and where certain master set entries cannot be deleted because they still have paths remaining to chains in one or more detail datasets.

19.3 PROGRESS REPORTING

In order to keep the user aware of how far through an archive ARCHIVE has processed it displays progress messages on $STDLIST. These messages are to be used as approximate guides only as it is not possible to estimate accurately the time required or number of entries which will be selected before they are actually archived. The name of either the specified data set, or one of its associated detail sets, will be displayed along with the number of entries which have been archived across for that set.

A percentage figure is shown which attempts to represent the percent complete for this dataset, but may not always be accurate - they are meant as a guide only. These messages are displayed every 10 seconds (unless specified otherwise - see Appendix C). In addition a file FXPROGRn (where n is a single digit 1-9) is created which contains the same information. Normally this file will be called FXPROGR1, but if another user is already using ARCHIVE, CREATE, ANALYZE or DIAGNOSE in the same logon group then the file will be FXPROGR2 or 3 etc.

```
<><><> ARCHIVE EXTRACTION COMPLETED <><><>
```

A new Source Database is then prompted for, enabling a completely new archiving procedure to begin.

19.4 ERROR AND WARNING MESSAGES

ERROR OPENING DATABASE
> This message is followed by the error message returned by the DBERROR intrinsic which explains why the database could not be opened.

ERROR - Invalid Database Name. Please re-enter

A database name cannot have more than a six character filename, and must start with a letter. The Source database name may be qualified with group and account names, which in turn must be valid.

ERROR - Extraction too complex for Classic HP and has been truncated

Programs running on an MPE/V machine, or in Compatibility Mode on an MPE/iX machine, are limited in the data stack size available. FLEXIBASE can run out of data stack space in these programs if it has to navigate through a long chain of associated data sets to extract data (i.e. copying one detail set entry can cause a whole series of associated data entries For ARCHIVE try and restrict the extraction to a subset of datasets, thereby restricting the complexity of the extract.

No such entry in database <<Source Database Name>>

When archiving entries you have specified a value in a particular set which does not exist in the Source database.

<<Dataset Name>> set is full - archiving incomplete

When archiving entries it is possible to fill up one of the associated datasets. If this occurs the data that has already been archived in this run is restored to the original source database, thus leaving it in a fully consistent state. The ARCHIVE procedure should be done again specifying a larger capacity for the dataset, or sets, concerned. A JCW is set when this condition occurs. The JCW name is SETFULL, and it contains the value 1 if any set in the ARCHIVE run fills up.

IMAGE Error at line number nnnn

Dataset: <<setname>> in Database: <<database name>>

An unexpected error has occurred while archiving data across. Should this message appear, contact Open Seas with all the details (including STDLISTs and any other relevant data).

[INVALID - Must select at least one dataset name to delete from]

When selecting a list of dataset names from which to delete archived entries the user has not selected any dataset names at all. At least one set name must be selected.

[INVALID - Must select at least one dataset name to terminate at]

When selecting a list of master dataset names at which to terminate the recursive archiving the user has not selected any dataset names at all. At least one master set name must be selected.

[INVALID - Only Master Sets may be selected]

Only Master sets may be selected as terminating sets as a Detail set must, by the rules of TurboIMAGE, also archive its' Master entries to be successfully put.

File exists with the same name as the new root file

There exists a file on the system with the same name as that requested for the Archive Database root file. Either choose another name, or purge this file if it is not needed.

WARNING ₵ Dataset filename conflict, cannot create database

Although the root file can be created there exist other file(s) with names the same as one or more of the Archive Database Datasets, e.g. NEWPAY03 etc. Either remove these files or choose another name for the Archive Database.

Destination Database Name may not be qualified

The Archive Database will be created in the logon group and account only, and so the name may not be qualified. If required, the Archive Database can be renamed upon completion using option 14 from the FLEXIBASE menu.

Entry with this key value already exists in <<sample database name>>

When archiving entries the user cannot specify a particular data item value in a specific set whose entry has already been archived across.

Data Item is not in set

The requested Data item is not defined as belonging to the specified Dataset.

Set does not exist in database
>   The requested Dataset is not defined as belonging to the specified Database.

Data item is not in database
>   The requested Data Item is not defined as belonging to the specified Database.

Schema file XPRQDASC can not successfully be put through DBSCHEMA
>   With a remote source database, ARCHIVE has to use a self-generated schema file XPRQDASC
>   which it then uses to create the destination database via use of DBSCHEMA.PUB.SYS. Any
>   errors in this schema file will cause this message to appear. It represents an unexpected error and
>   should be reported to your local Open Seas office. If an unexpected program error occurs, such as
>   a bounds error etc., the program will abort with the following type of message:

<><> Program Error Occurred in Module ARCHIVEBYVALUE <><>
STRING INDEX EXCEEDS CURRENT LENGTH (PASCERR 651)
Program Module Trace:
   OUTERBLOCK
   ARCHIVEBYVALUE
Stack Marker:
   Index Register %000000 (Decimal   0)
   Code Location   %004550 (Decimal   2408)
   Status   %060004 (Decimal 24580)
   Delta-Q   %000027 (Decimal   23)

>   To report this fault please send the following to Open Seas:
>   1) Current Schema File of Database,
>   and
>   2) This printout
>   <><><><><><><><><><><><><><><><><><><><><><><><>

Such an abort has not been expected, and as such represents a problem either with the software or database
itself. It will require further investigation, for which purpose the items listed above should be sent to Open
Seas.

[Invalid option] or [INVALID RESPONSE PLEASE RE-ENTER]
>   There are some questions where there is an allowable range of acceptable answers (e.g., Y or N).
>   If an invalid option has been entered this message is displayed and the user must re-enter one of
>   the allowed values.

Error in << DB Call>> on Database "dname", Set Number "n"
>   ARCHIVE uses many standard IMAGE calls when operating. Any error encountered in these
>   calls will result in this message (should never occur unless there is some problem with the
>   database). Contact Open Seas if this error occurs.

When specifying data value conditions a number of errors may occur. In each case ARCHIVE will display
the error and attempt to locate the error on the input line by means of a "^" symbol positioned under the
offending portion.

The following errors may be encountered:

[INVALID Condition - Sub Item is faulty]
>   The number entered in parentheses after the data item name is invalid (perhaps greater than the
>   maximum number of subitems.

[INVALID Condition - Relational Operator is Faulty]
>   Only the relational operators "=", ">", "<", "<>", ">=" or "<=" are allowed to follow the data item
>   name, and MUST precede the item value.

[INVALID Condition - Item Value is Faulty]

Perhaps a non-numeric value has been entered for a numeric item?

[INVALID Condition - Item Name is Faulty]
The specified item name does not exist in the specified data set.

[INVALID Condition - Wildcards may only be used with "="]
The wildcard character "@" may only be used in conjunction with the relational operator "=".

[INVALID Condition - An AND or an OR is incomplete]
A data item name, relational operator and value has been specified followed by an AND and an OR ₵ ARCHIVE now expects another condition to follow. Alternatively the user has not entered AND or OR correctly.

[INVALID Condition - A Maximum of 3 item conditions are allowed]
Three conditions only may be entered separated by AND/OR.

When entering the capacity percentage or formula the following error messages may be encountered:

[INAVLID Formula - missing "*"]
When specifying a multiple of the entry count the user MUST enter ENT followed by the character "*" (indicating "multiplied by"), and then a number.

[INVALID Formula - percentage faulty]
The percentage entered MUST be in the range 0 to 100 for capacities, and must be non-zero, or 0 to 200 for entries, and again must be non-zero.

SECTION 20 C MPE FUNCTION

The purpose of the MPE Commands function on the main FLEXIBASE menu is to allow the user to enter MPE commands or run other programs without having to leave the FLEXIBASE program. All FLEXIBASE functions return to the main menu upon successful completion, including the MPE Commands function.

Any MPE command that is programmatically executable (see the MPE Commands Reference manual) may be entered, plus any legitimate parameters of those commands. UDC commands may not be entered however.

In addition FLEXIBASE supports the use of the RUN command to execute non-FLEXIBASE programs such as QUERY, EDITOR etc. All permitted parameters to the run command may also be entered. To ease the execution of programs which reside in PUB.SYS the user may simply enter the name of the program. For example, to run QUERY.PUB.SYS, enter the name QUERY and FLEXIBASE will run the program for you.

To return to the main FLEXIBASE menu enter a carriage return in response to the prompt, otherwise another MPE command may be entered.

SECTION 21 Ȼ SPLITTING AND JOINING SETS / ITEMS

Using combinations of the UNLOAD, TRANSMUTE and LOAD functions it is possible to split and join (i.e. merge) datasets and data items as shown in the following example. These procedures make use of the fact that data unloaded to an MPE file is independent of item boundaries and is simply a series of contiguous bytes.

Example A Ȼ   SPLIT AN X10 ITEM INTO TWO ITEMS OF X4 AND X6

- UNLOAD the set containing the item into an MPE file.
- ERASE the set.
- TRANSMUTE the database redefining the item as two items in the item list and in the relevant set but keeping the two items contiguous in the set and in the same position within the set as the old item. This will be a quick run since there are no actual entries to be transformed. The new entries must have a combined word length equal to the word length of the old item.
- LOAD the MPE file back into the set.
- CHECK that the data is as required using QUERY.

Example B Ȼ   SPLIT A SIMPLE X24 ITEM INTO A COMPOUND X6 ITEM

- Proceed as in Example A but redefine the item as a compound item with 4 occurrences of X6.

Example C Ȼ   JOIN AN X4 AND X6 ITEM INTO A SINGLE X10 ITEM

- TRANSMUTE the database if necessary to ensure that the two items are contiguous and in the required sequence in the relevant set.
- UNLOAD the set containing the items into an MPE file.
- ERASE the set.
- TRANSMUTE the database, redefining the two items as a single item in the item list and relevant set but keeping the new item in the same position within the set as the old item. The new item must have a word length equal to the sum of the word lengths of the old items.
- LOAD back the MPE file into the set.
- Check that the data is as required using QUERY.

Example D Ȼ   MERGE A COMPOUND X6 ITEM INTO A SIMPLE X24 ITEM

- Proceed as in Example C but redefine the item as a simple item.

Example E Ȼ   SPLIT A SET INTO TWO SETS

- Back-up the original database using DBSTORE/STORE.
- TRANSMUTE the database redefining the old set as the first of the new sets in the schema.
- UNLOAD the first new set to an MPE file.
- RESTORE the back-up copy of the database.
- TRANSMUTE the database redefining the old set as the second of the new sets in the schema and including the first of the new sets in the schema Ȼ the latter will be created empty.
- LOAD the first of the new sets using the MPE file created in Step 3.
- CHECK that the data is as required using QUERY.

Example F ₵ JOINING TWO SETS INTO ONE SET

- UNLOAD the first of the sets to an MPE file.
- UNLOAD the second of the sets to an MPE file.
- ERASE both sets.
- TRANSMUTE the database redefining the two old sets as one new set. This will be a quick run since there are no actual entries to be transformed.
- Process the two MPE files to produce a single file which matches the new set definition, using EDITOR or other utility programs.
- LOAD the new set using the MPE file produced in Step 5.
- Check that the data is as required using QUERY.

SECTION 22 C DATA BASE ENQUIRY AND REPORTING

INTRODUCTION

An optional facility with FLEXIBASE is the FANTASIA REPORTER software. The FANTASIA REPORTER is a reporting system for HP3000 or TurboIMAGE data bases. It consists of two basic modules:

- A user friendly, menu-based module to provide easy ad-hoc enquiry and reporting from IMAGE databases. This module is named QUIP which stands for Q-GEN User Interface Program. This generates interpretive report procedures, or if you have a COBOL compiler, compiled object programs via the Q-GEN Compiler.
- A second module, named the Q-GEN Compiler, for converting interpretive procedures to compiled object programs.

Because the FANTASIA REPORTER can create optimized compiled COBOL report programs (instead of using the interpretive approach of most other report writers) it is much more efficient at run time.

The QUIP module also contains a flexible data dictionary that you can use to document your databases. It simplifies use of the report writer because you can pre-configure item edits, database access and other information into the dictionary.

An interface is also provided to the FANTASIA LASER SERVER software to provide enhanced, typeset report printing and automatic forms overlays on HP LaserJets.

Note that this Section of the FLEXIBASE manual provides only a very brief description of the capabilities and facilities of the FANTASIA REPORTER software. It is more fully documented in a separate manual.

PRODUCT STRUCTURE

Figure A shows the inter-relationship between the FANTASIA REPORTER modules and displays the various paths that can be taken to produce a report.

As can be seen, there is great flexibility provided. For the unsophisticated user the whole report production process can be seen as a single step from a simple menu to report output. However for the expert user intervention at Command Language or COBOL source code level is available.
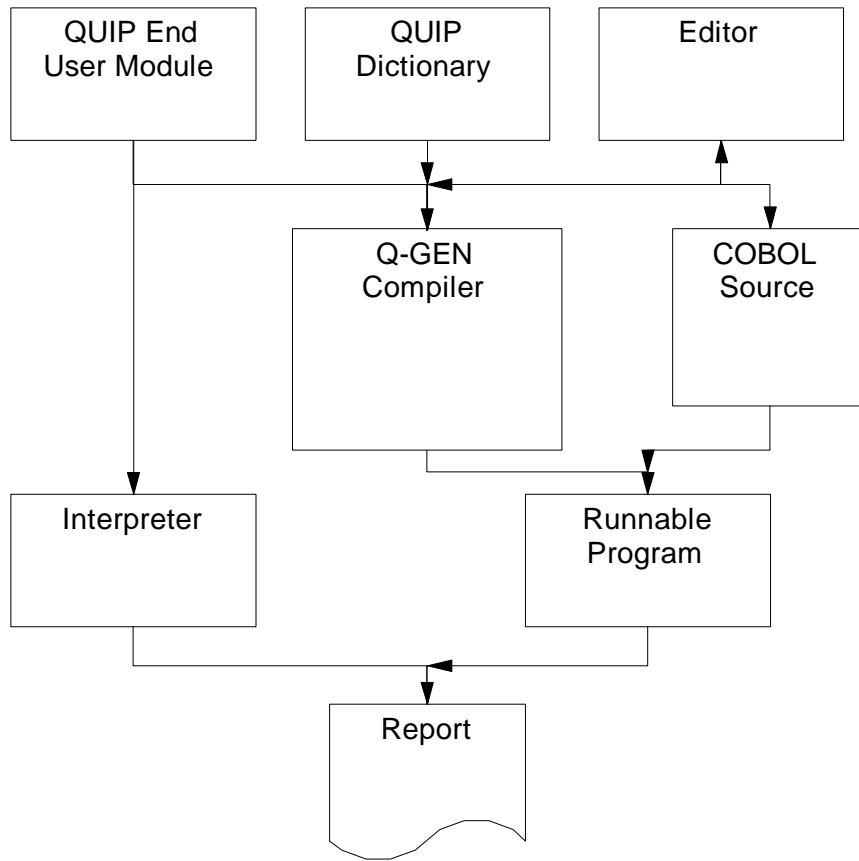

Figure A FANTASIA REPORTER Modules

In addition, because the Q-GEN Compiler Command Language is an enhanced and extended version of the HP QUERY language, the Q-GEN Compiler can also convert any existing QUERY report procedures into equivalent COBOL programs. This reduces the run time of such reports enormously.

It can also be seen from *Figure A* that the QUIP module can be employed as an easy-to-use menu-driven reporting system even if the Q-GEN Compiler is not purchased. However use of the Compiler provides additional facilities and faster run times.

PERFORMANCE

On a simple report, a program generated by the FANTASIA REPORTER is likely to run in about half the time (both in terms of CPU seconds and elapsed time) than an equivalent interpreted procedure. On a more complex report using sorting and totalling, it will be up to 3 times faster. On a report that retrieves data from several data sets (e.g. using the JOIN/MULTIFIND commands of QUERY) it will be up to 8 times faster. These performance figures have been verified by a series of tests on various HP3000 models (both "Classic" and Precision Architecture models). The user manual contains more detailed information on performance comparisons.

The performance advantages are obtained primarily because of the superiority of well written compiled COBOL code over the interpreted process of other systems. There are also fewer disc accesses because of the design of the Q-GEN Compiler (as an example MANUAL MASTERS are only read when there is a change of key value). The performance of Q-GEN Compiler generated report programs is superior to all other commonly used report writers on HP3000 computers, including QUIZ (from COGNOS) and Business Report Writer (from Hewlett Packard).

SAMPLE PERFORMANCE DATA

Typical Interpreter (HP QUERY)

95 Minutes

Q-GEN

13 Minutes

Report Generated from 2 data sets of greater than 25,000
records with sorting and totalling at 2 levels
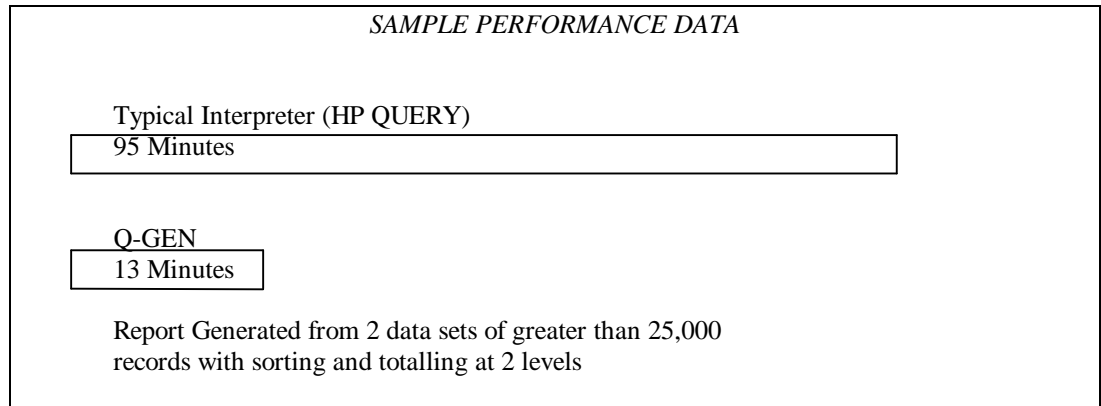
FAST MODULE

For users still running on MPE/V systems, there is a module supplied that can improve performance even further by replacing IMAGE calls by direct file system calls within the programs generated by the Q-GEN Compiler. Use of this module can reduce the run time of reports that perform serial reads so that they are up to twice as fast as standard Q GEN Compiler generated report programs. The fast read facility can also be used to improve the performance of user written programs.

THE QUIP MODULE

The Q-GEN User Interface Program (QUIP Module) is designed to enable untutored end-users to retrieve information from IMAGE databases and format it on their terminal or output it to a printer.

The user is prompted (by the display of menus) for such information as data selection criteria, data items to be printed, report headings etc. Unlike other similar products it has three major advantages:

- Very high performance due to the use of the Q-GEN Compiler and optimized design.

- The system manager can easily control the size of database scans and also restrict usage in other ways (e.g. can force certain reports to be streamed as jobs if initiated by certain users).
- Data dictionary use is optional thus reducing the cost and complexity involved in installation and use of the product.

QUIP MENU LEVELS

```
┌─────────────────────────────────────────────────────────┐
│            ┌───────────────────────────────────┐         │
│            │       CHOOSE DATA BASES            │         │
│            └───────────────────────────────────┘         │
│            ┌───────────────────────────────────┐         │
│            │  SPECIFY REPORTED AND SORT ITEMS   │         │
│            └───────────────────────────────────┘         │
│            ┌───────────────────────────────────┐         │
│            │       SPECIFY CALCULATIONS         │         │
│            └───────────────────────────────────┘         │
│            ┌───────────────────────────────────┐         │
│            │     SPECIFY RECORD SELECTION       │         │
│            └───────────────────────────────────┘         │
│            ┌───────────────────────────────────┐         │
│            │     SPECIFY EDITS AND TOTALS       │         │
│            └───────────────────────────────────┘         │
│            ┌───────────────────────────────────┐         │
│            │     SPECIFY TITLES AND FORMAT      │         │
│            └───────────────────────────────────┘         │
│            ┌───────────────────────────────────┐         │
│            │   CHOOSE MODE AND OUTPUT DEVICE    │         │
│            └───────────────────────────────────┘         │
│            ┌─────────────────┐  ┌───────────────┐         │
│            │    TERMINAL     │  │    PRINTER    │         │
│            └─────────────────┘  └───────────────┘         │
└─────────────────────────────────────────────────────────┘
```
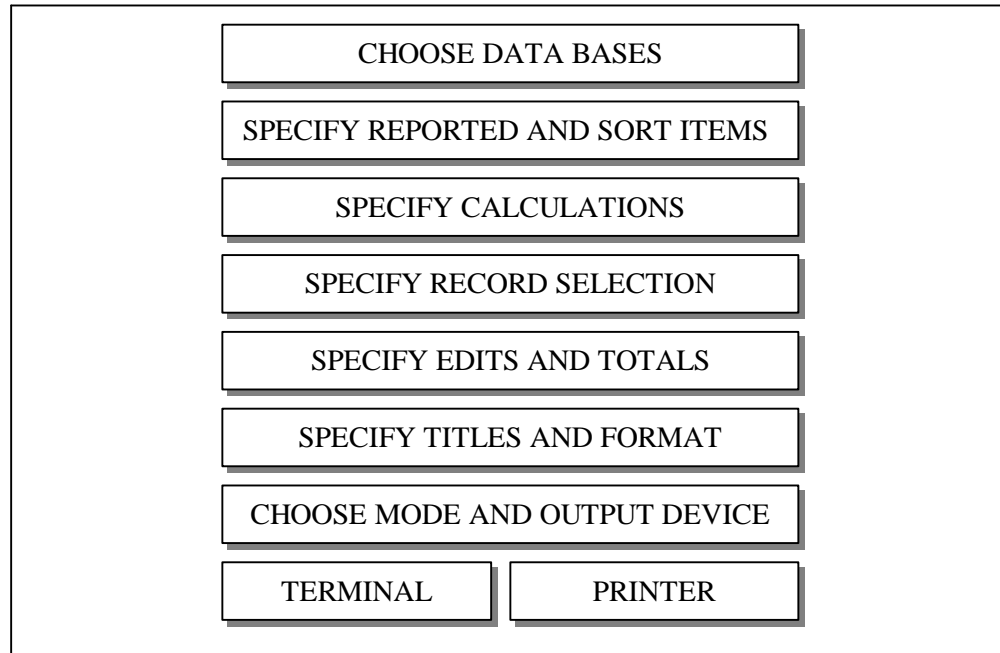
The QUIP module does not require knowledge of IMAGE, or any other expertise. However it can also be used by software staff to produce skeleton Command Language procedures or COBOL programs for further refinement.

THE QUIP DICTIONARY

The QUIP Dictionary provides the system administrator with the means to control system resources available to QUIP users. Additionally the QUIP Dictionary can also be used to improve the QUIP/User Interface. Features of the Dictionary include:

- Control of maximum data set scans on a Global and/or a User level.
- Restricting database access for a user and/or a group of users.
- Providing data set and data item aliases and descriptions on QUIP menu screens.
- Providing default data item edits and column headings on end-user reports.
- Routing printer output to local printers.

There are many other features in the Dictionary, which is maintained by an easy-to-use V-Plus based program.

THE Q-GEN COMPILER

The Q-GEN Compiler will produce a runable object program given only the name of a report procedure written in the Command Language or HP QUERY language.

Various optional control commands can also be entered, e.g. to indicate that a COBOL source program is required. No programming skills are required to use the Q-GEN Compiler and minimal training is necessary to use the Command Language.

COBOL PROGRAM GENERATION

Because one of the options in the Q-GEN Compiler is to generate a COBOL source program, it is possible to produce reports that could not otherwise be done with any other reporting system. For example if you require a very complex algorithm to be performed on the retrieved data you can generate a skeleton COBOL program and then amend it.

The source code that the FANTASIA REPORTER produces has been specifically designed to make amendment easy. It is good quality, well structured code. Note that the FANTASIA REPORTER will not only generate report printing programs but will also produce on-line enquiry programs. It can therefore be used as a general program generator to assist in the development phases of new computer systems.

Once generated the COBOL programs can be freely moved to additional processors without the need for a run-time license.

FLEXIBILITY

The FANTASIA REPORTER supports many facilities not in other report writers which makes it easier to specify more complex reports. Among the features are the following:

- Conditional printing, totalling or arithmetic using an IF statement.
- Finding, sorting or printing using part of a data item.
- Easy transfer of totals or other values from one report to another where several reports are required from the same data.
- European date format, as an option.
- Easy and efficient retrieval of data from multiple data sets.
- Optional output to disc for downloading to PCs (e.g. DIF format).

LASER PRINTING INTERFACE

The FANTASIA REPORTER has an in-built interface to the FANTASIA LASER SERVER software. The FANTASIA LASER SERVER software is another software product produced by Open Seas that supports sophisticated use of HP LaserJet printers. By selecting the "FANTASIA" output option in the QUIP module you can select which fonts to use for page headings, column headings and detail report lines. The software will then automatically build a suitable form incorporating those specifications so that the output looks much better than most computer printed reports.

In addition you can design a form to overlay onto the data (either with FANTASIA/3000 commands or the facilities of FANTASIA PC) that is automatically invoked by an interpretive QUIP procedure or a Q-GEN compiled program.

USER INTERFACE

The normal mode of use of the FANTASIA REPORTER is via the QUIP menus. The instructions on the following pages explain how to run the QUIP menu program. Note that it uses V-PLUS so you normally have to press Enter rather than Return to enter information (or use the labelled function keys where appropriate).

We recommend compilation in almost all cases (assuming that you have a COBOL compiler) particularly when running on MPE/iX.

RUNNING QUIP

QUIP can be run in session mode by entering:

```
:RUN QUIP.PUB.HQ; or :RUN QUIPX.PUB.HQ for the native mode version
on MPE/iX.
```

Initially the program banner screen is displayed. The ENTER key must be pressed to see the main selection menu. The contents of the screen are dependent on the configuration options set by your system manager but typically will appear as below.
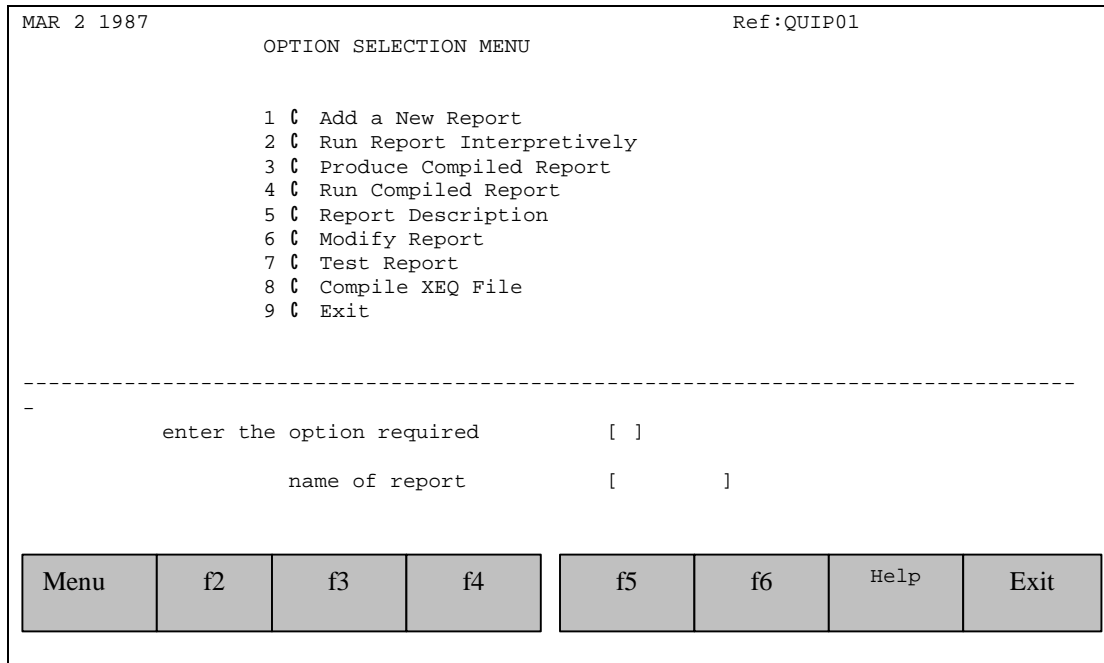
```
MAR 2 1987                                              Ref:QUIP01
                   OPTION SELECTION MENU


                   1 ℂ  Add a New Report
                   2 ℂ  Run Report Interpretively
                   3 ℂ  Produce Compiled Report
                   4 ℂ  Run Compiled Report
                   5 ℂ  Report Description
                   6 ℂ  Modify Report
                   7 ℂ  Test Report
                   8 ℂ  Compile XEQ File
                   9 ℂ  Exit



----------------------------------------------------------------------------
-
          enter the option required         [ ]

                  name of report            [       ]
```

Menu	f2	f3	f4	f5	f6	Help	Exit

FIG. 2-1 Main Selection Menu

Each option is described in detail below:

OPTION 1   ADD A REPORT. Selection of this option will take the user through the steps of creating a new report.

OPTION 2   RUN REPORT INTERPRETIVELY. This option causes a report that has already been defined using QUIP to be executed interpretively. It provides the user with a fast response for simple ad-hoc reports that are run infrequently. However if the report is complicated or will be run on a regular basis the report should be compiled as this will be more efficient in terms of the run time and machine resources used. Note that QUIP generates QUERY statements in this case and programmatically executes them via the HP QUERY interpreter.

OPTION 3   COMPILE REPORT. This option will cause the QUIP report to be compiled into an object program (and optionally into COBOL source code) by means of the Q-GEN Compiler. A compiled program is the most efficient method of running a complex or frequently used report.

OPTION 4   RUN COMPILED REPORT. This option will run a report that has previously been compiled using Option 3.

OPTION 5   REPORT DESCRIPTION. This option provides a brief description of a QUIP report. The output can either be to the terminal or to the system printer.

OPTION 6   MODIFY REPORT. This option allows a report defined using option 1 to be modified.

OPTION 7   TEST REPORT. This option runs a report interpretively using a subset of the data available. Allows the format of the report to be viewed without retrieving all the data.

OPTION 8   COMPILE XEQ FILE. This option compiles an XEQ file generated by options 2 or 7. Note if the XEQ file was generated by option 7 a subset of data will be selected.

OPTION 9   EXIT. This option will stop execution of QUIP and return you to command level. You may also use the f8 function key at any time to stop execution of QUIP.

Field Descriptions

Option No.   Enter the number of the option you require.

Report Name   The name of the report to be actioned. It must be entered for all options except when exiting QUIP.

Pressing f7 (Help) will display a help screen.

When adding a report, if a file already exists with the same report name, you may either enter a new name or overwrite the existing report by pressing the f2 (Overwrite) function key.

OPTION 1 ADDING A REPORT

Database Selection Screen

This screen is used to select the databases to be accessed by QUIP. A maximum of 16 databases can be accessed within a single report.

The screen will appear as shown below if no default databases have been defined.

```
    Define the database/s to be accessed : -

      Data base name [     ]  Password [       ]      Mode [ ]

                               Group    [      ]  Account [       ]
--------------------------------------------------------------------------------
   -
```

Menu	f2	Select Items			f6	f7	Exit

FIG. 2-2 Database Selection Screen

Field Descriptions

Database name	The name of the database to be accessed.
Password	The database password for the above database. Note that this must be entered in the correct case as with all IMAGE passwords.
Mode	The database access mode for the above database. The default value for this field is 5.
Group	The name of the group in which the above database resides. It is only required if it is not in the log-on group.
Account	The account in which the above database resides if it is not in the log-on account. Note that if this is used then the Group name must also be entered.

The format of the database selection screen shown hereafter is typical when default databases have been defined.

```
    Define the database/s to be accessed : -

      Data base name [STORE ]  Password [******]     Mode [5]

                              Group    [PUB  ]  Account [HQ     ]

----------------------------------------------------------------------------------
-
                    ACCESSIBLE DATA BASES
      Base    Group      Account
  1   STORE   PUB         HQ
       Data base used in the Q-GEN demonstration system
  2   QUIPDB  PUB         HQ
       QUIP configuration data base
  3   SLEDG   SALES       LEDGER
       Sales ledger data base
```

Menu	f2	Select Items	Prev Page	Next Page	f6	f7	Exit

FIG. 2-3 Database Selection Screen

With this screen there are 3 ways to select a database:

- The database definition field already contains the first of the default databases; to access this press enter.
- To access one of the other default databases displayed on the screen enter the index that is beside the database in the Data Base Name field and press return.
- To access a database other than one of the default databases enter the details as described on the previous page.

There may be further databases that you can access. These can be seen by pressing f5 (next page). Pressing f4 (prev page) will display the previous 6 databases.

Data Set Selection Screen

This screen is used to select the data sets (i.e. files) to be accessed in the database that was last specified. The screen will be appended to the base selection screen. After making your choice of data sets you will be returned to the database selection screen. The middle section of the screen contains the names and the index numbers of the sets available in the database.

```
       Define the database/s to be accessed : -

         Data base name [     ]  Password [      ]      Mode [ ]

                                  Group    [     ]  Account [       ]

--------------------------------------------------------------------------------
-

     1   SALES-MSTR        2   PURCHASE-MSTR        3   STOCK-MSTR
     4   SALES-HEADER      5   SALES-LINES          6   PURCHASE-HEADER
     7   PURCHASE-LINES




--------------------------------------------------------------------------------
-

     enter numbers of datasets to be
          accessed : [                                              ]
```

Menu	f2	Select Items	Prev Page	Next Page	f6	f7	Exit

FIG. 2-4 Data Set Selection Screen

Field Descriptions

Dataset to be accessed: The index numbers of the data sets to be selected. If more than one dataset is selected the index numbers must be separated by commas or spaces. The numbers must be entered in the order you want them to be accessed in the report.

Note that this screen is typical of all the screens where multiple values can be entered, in that the values must be separated by commas or spaces.

Pressing f7 (Help) function key will take you into the help system where you can get help on any of the displayed data sets.

Item Selection Screen

This screen is used to select the data items to be:
- reported in detail lines of the report
- used for selection of the data
- on which the report is to be sorted

The top section of the screen contains the names and the index numbers of the items available. If there are more items available than will fit on a single screen, you may page through the items using the next (f4) and previous (f5) page function keys.

```
MAR 2 1987                                           Ref:QUIP20
                      AVAILABLE DATA ITEMS

   Items Available for Set :- PURCHASE-MSTR
   1  PUR-ORDER-NO     2   SUPPLIER-CODE      3  ORDER-DATE
   4  DELIVERY-DATE    5   PUR-ORDER-NO       6  STOCK-CODE
   7  ORDER-QTY        8   ITEM-PRICE         9  LINE-NO (4)




----------------------------------------------------------------------------------
-

     enter numbers of items to be:-

       reported ............ [                                   ]
       selected on ......... [                                   ]
       sorted upon ......... [                                   ]
       subtotals(Y/N)........[Y]  edits for numeric items(Y/N) [Y]
```

Menu	f2	Define Calc	Prev Page	Next Page	f6	f7	Exit

FIG. 2-5 Item Selection Screen

Field Descriptions

Reported — The index numbers of the displayed items to appear on the report as detail items. The index numbers must be separated by commas.

Selected on — The index numbers of the displayed items that will be selected by value during the record retrieval phase of the report.

Sorted — The index numbers of the displayed items upon which the report will be sorted. The items will normally be sorted in ascending sequence. If the items are to be sorted in descending sequence the index number must be followed with a "D".

Example: [1,2D,4   ]

Subtotals — This option allows the user to define subtotals for the reported numeric items. Default value is Y for Yes.

Edits for numeric items — This option allows the user to define output edits on the reported numeric items. Default value is Y.

It is possible that you may want to define a calculation based on one or more of the displayed items. To do this press the f3 function key, this will display the calculation definition screen. Once defined a calculation can be like any other item except it cannot be used in a selection nor used as a sorted item.

There may be more items available than can fit onto one screen. You can move between pages by using the next page (f5) and previous page (f4) function keys, which will display the next and previous screen of items respectively.

If an item name is followed by a number in brackets then there is more than one occurrence of that item. The number of occurrences is specified by the number in the brackets. To report these items a subscript must be entered after the item number, e.g. to report the second occurrence of LINE-NO in the above screen enter 9(2). If the first occurrence is required there is no need to use a subscript.

Calculation Definition Screen

This screen is used to define calculations to be used in the report. A calculation may be based on the values of Numeric Items displayed on the upper part of the screen, or on numeric values.

```
MAR 2 1987                                      Ref:QUIP20
                      AVAILABLE DATA ITEMS


   Items Available for Set :- PURCHASE-MSTR
   1  PUR-ORDER-NO      2   SUPPLIER-CODE      3   ORDER-DATE
   4  DELIVERY-DATE     5   PUR-ORDER-NO       6   STOCK-CODE
   7  ORDER-QTY         8   ITEM-PRICE         9   LINE-NO




   -------------------------------------------------------------------------------
   -
               CALCULATION DEFINITION              Ref:QUIP28
   Calculation Name : [Total-Line-Value]
                       Operator                    Operator
   Value1:[#8              ] [MUL ] Value2:[#7          ] [    ]
   Value3:[                ] [    ] Value4:[            ] [    ]
   Value5:[                ] [    ] Value6:[            ] [    ]
```

Menu	f2	f3	Prev Page	Next Page	f6	Help	Exit

FIG. 2-6 Calculation Definition Screen

Field Descriptions

Calculation Name	This is the name given to the calculation. It may contain Upper and Lower case letters as well as numeric digits. Use a name that best describes the calculation. The calculation name will appear on the report as a column. Some examples of calculation names are:

> Total Cost   Total-Cost
> First Value   1st Value

Value1 etc	These are the values to be used in the calculation. A value can be an available item displayed on the upper part of the screen or a numeric value. If the value is an available item it must be prefixed with a hash (#). Some examples of valid value entries are:

#7 (specifies item number 7 of the available items)
100 (specifies a numeric value of 100)
10.5 (specifies a numeric value of 10.5)

If there is more than one occurrence of the item (this is indicated when the item is followed by a number in brackets e.g. LINE-NO(4)), the nth occurrence of the item can be used in the calculation by using a subscript after the item number e.g. #7(2).

Operator   This is the arithmetic operator that is applied to the values. The valid operators are:

> ADD ₵ Add
> SUB ₵ Subtract
> MUL ₵ Multiply
> DIV ₵ Divide

Care should be taken in the order that the values and operators are entered, as the calculation is performed sequentially. An example is "find 10% of 110". This can be entered in two ways:

> 110 DIV 100 MUL 10
> or
> 110 MUL 10 DIV 100

These will produce different results (the first gives the result of 10 and the second 11).

To return to the item selection screen press f1 (Exit Calc) function key.

Pressing the f7 (help) function key will display a help screen.

Selection Definition Screen

This screen is used to define the selection criteria for the report. The user is prompted for selection information for "selected on" items specified on the previous ITEM SELECTION SCREEN.

```
MAR 2 1987                                              Ref:QUIP20
                         AVAILABLE DATA ITEMS

   Items Available for Set :- PURCHASE-MSTR
   1  PUR-ORDER-NO    2  SUPPLIER-CODE       3  ORDER-DATE
   4  DELIVERY-DATE   5  PUR-ORDER-NO        6  STOCK-CODE
   7  ORDER-QTY       8  ITEM-PRICE          9  LINE-NO




----------------------------------------------------------------------------
-
     enter selection criteria for : SUPPLIER-CODE

        relationship with last selection [   ]

        selection operator... [ ]

        selection value.......[                                    ]
```

Menu	f2	Reselec	Prev Page	Next Page	f6	f7	Exit

FIG. 2-7 Selection Definition Screen

Field Descriptions

Relationship with last selection   The relationship with the previous selection. This field will not appear for the first selection to be defined. The relationship can be either "AND" or "OR".

Selection operator   This is the relationship between the item being defined and the selection value. The operator can be one of :

```
OPERATOR              MEANING
   EQ                 is equal to
   NE                 is not equal to
   LT                 is less than
   LE                 is less than or is equal to
   GT                 is greater than
   GE                 is greater than or is equal
   MA                 matches pattern
```

Selection Value   The value to be used for selection. The MA operator will cause the report to perform a generic search. This is typically used to find an item that matches a pattern of characters. In this case the selection value must contain a MATCH PATTERN for example:

SON@   will find all occurrences of the item beginning with SON e.g. SONYEK.

@SON   will find all occurrences of the item ending with SON e.g. WILSON.

@SON@   will find all occurrences of SON anywhere in the item e.g.
            SONYEK, WILSON, MYSONBY.

The MA operator can only be used on the first selection and in compiled
            reports.

The selection value can be entered when the report is run. In order to do
            this enter "" (two double quotes) in the selection value field.

To select an item that contains only spaces the keyword SPACES must
            be entered in the selection value.

Edit Definition Screen

This screen is used to define output edits for numeric items. It will be displayed if you have requested edits and have selected numeric items for reporting on the ITEM SELECTION SCREEN. An edit can be used to change the default representation of numeric items to improve the appearance and readability of your reports.

QUIP will prompt for an edit for each reported numeric item. If an edit is not required for a particular item, skip to the next item by using the f5 function key. You may copy the previous edit by pressing the f4 function key.

```
MAR 2 1987                                         Ref:QUIP20
                      AVAILABLE DATA ITEMS

  Items Available for Set :- PURCHASE-MSTR
  1   PUR-ORDER-NO     2    SUPPLIER-CODE     3   ORDER-DATE
  4   DELIVERY-DATE    5    PUR-ORDER-NO      6   STOCK-CODE
  7   ORDER-QTY        8    ITEM-PRICE        9   LINE-NO








--------------------------------------------------------------------------------
-
     define edit mask for item : SUPPLIER-CODE

    number of decimal places..........................[ ] (0/9)
    number of digits before decimal point.............[ ] (0/18)
    Optionally
       replace leading zeros with spaces or *.........[ ] (S or *)
       replace left most zero with $ (dollar sign)....[ ] (Y or N)
       replace negative sign with CR..................[ ] (Y or N)
```

Menu	f2	Re-select	Copy Last	Skip to Next	f6	f 7	Exit

FIG. 2-8 Edit Definition Screen

Field Descriptions

Number of decimal places   The number of decimal places to be displayed. A decimal point will be inserted in front of the decimal digits. A value in the range 0 to 9 must be entered. If 0 is entered then a decimal point will not be printed.

Digits before decimal point   The number of digits to be displayed in front of the decimal point. A value in the range 0 to 18 must be entered. If 0 is entered then no digits will be displayed in front of the decimal point.

Replace leading zero/s   Any leading zeros can be replaced with spaces or asterisks. To replace leading zeros with spaces enter S. To replace with asterisks enter *. Leaving the field blank will cause leading zeros to be displayed.

Replace left most zero with dollar   This will replace the left most (leading) zero with a dollar ($) sign. If there is not a leading zero then the dollar sign is displayed in front of the left most digit. A Y or N reply must be entered in this field.

Replace negative with CR   This option will replace the negative sign with a credit symbol (CR). A Y or N reply must be entered in this field.

Subtotals Definition Screen

This screen is used to define output subtotals for numeric items. It is displayed if you have requested subtotals and have selected numeric items for reporting on the ITEM SELECTION SCREEN. A subtotal for a numeric item will be printed whenever a sorted item changes value.

QUIP will prompt for a subtotal for each numeric item reported, for each sort level.

```
MAR 2 1987                                                Ref:QUIP20
                          AVAILABLE DATA ITEMS

    Items Available for Set :- PURCHASE-MSTR
    1  PUR-ORDER-NO      2    SUPPLIER-CODE      3   ORDER-DATE
    4  DELIVERY-DATE     5    PUR-ORDER-NO       6   STOCK-CODE
    7  ORDER-QTY         8    ITEM-PRICE         9   LINE-NO








-------------------------------------------------------------------------------
-
     define subtotals for sorted item : PUR-ORDER-NO

        subtotal calculated for : ORDER-QTY
        subtotal to be based on :
                    Accumulation......................[ ]
                 or Average...........................[ ]
                 or Count.............................[ ]
```

Menu	f2	Re-select	f4	Skip to Next	Next Level		Exit

FIG. 2-9 Subtotals Definition Screen

Field Descriptions

Accumulation	This accumulates the value of the item since the previous sort break, i.e. provides a simple total.
Average	This causes the average value of the item since the previous sort break to be calculated and reported.
Count	This causes the number of values since the previous sort break to be counted and reported.
NOTE:	If Subtotals are not required for the displayed sorted item, press the Next Level (f6) function key. The program will then go to the next sorted item.

Report Title Screen

This screen is used to enter the report title. The report title will be printed at the top of each report page.

```
MAR 2 1987                                              Ref:QUIP20
                        AVAILABLE DATA ITEMS

   Items Available for Set :- PURCHASE-MSTR
   1  PUR-ORDER-NO     2   SUPPLIER-CODE      3   ORDER-DATE
   4  DELIVERY-DATE    5   PUR-ORDER-NO       6   STOCK-CODE
   7  ORDER-QTY        8   ITEM-PRICE         9   LINE-NO








   ------------------------------------------------------------------------------------
   -

        Enter report title .......[                               ]
        Compress or Spread items..[C]   (C/S)
        Summary Totals only.......[N]
        Pause at Page Break.......[Y]
```

Menu	f2	Re-select	f4	f5	f6	f7	Exit

FIG. 2-10 Report Title Screen

Field Descriptions

Report Title	This is a 50 character free format field. There is no need to center the title in the field as this is automatically performed by the report generator.
Compress or Spread	This option controls the format of the reported items. Compressed format will print the items with a single space between each item. Spread format will space the items evenly across the page. The default value is Compressed (C).
Summary Totals	If Y is entered for this option then only the total lines will be printed (i.e. detail lines are suppressed). The default value is N (i.e. detail lines are printed).
Pause at Page Breaks	If Y is entered for this option then then report will pause at the bottom of each page. The default value is Y.

OPTION 2 RUNNING A REPORT INTERPRETIVELY

Interpretive Parameters Screen

This screen is used to enter the parameters for running a report interpretively.

```
MAR 2 1987                                               Ref:QUIP30

   Run report interpretively C   QUIPREP

   Report to be :-
      Run in job or interactively(J/I)........[ ]
      Output to :-
        Terminal(T), System Printer(P)
        Disc(D), Attached Printer(A)..........[ ]

   Query XEQ file to be generated :-
      Name of Query XEQ file.................[        ]




   ---------------------------------------------------------------------------
   -
```

Menu		f3	Report Descr.	f5	f6	f7	Exit

FIG. 2-11 Interpretive Run Parameters

Field Descriptions

Run in job or interactively	This will select the mode that the report will run in. If job (J) is selected the report will be scheduled as a job. If the report is selected to run interactively, it will run from the terminal. Any selection made by the user may be overridden by the system administrator.
Output to	This controls where the report will go. Output to a printer will be 132 characters wide. Output to the terminal will be 79 characters wide. Output to disc is 128 characters. The width of an attached printer is specified on a following screen.
Name of Query XEQ file	This is the name of the QUERY report file that will be generated. QUIP will actually generate 3 files: a database definition file (this will be suffixed with a D); a find definition file (suffixed with F); and finally a report definition file (suffixed with R).

Parameters for Attached Printer

This screen is used to enter the parameters for an attached printer.

```
MAR 2 1987                                              Ref:QUIP30

  Run report interpretively C   QUIPREP

  Report to be :-
     Run in job or interactively(J/I)........[ ]
     Output to :-
       Terminal(T), System Printer(P)
       Disc(D), Attached Printer(A)..........[ ]

  Query XEQ file to be generated :-
     Name of Query XEQ file.................[       ]




-------------------------------------------------------------------------------
-
              PARAMETERS FOR ATTACHED PRINTER

    Number of Columns (characters across page).............[80]

    Number of Lines per page..............................[66]

    Pause at page breaks..................................[N]
```

Menu	f2	f3	f4	f5	f6	f7	Exit

FIG. 2-12 Attached Printer Parameters

Field Descriptions

Number of Columns   This specifies the number of columns or characters that will be printed across the page (maximum is 132, minimum is 60).

Number of Lines   This specifies the number of lines or rows that are printed between page breaks (maximum is 66, minimum is 22).

Pause at Page Breaks   If Y is entered for this option the report pauses at page breaks. Use this option if your printer uses single sheet paper.

Parameters for Disc Output

This screen is used to enter the parameters for output to a disc file.

```
MAR 2 1987                                              Ref:QUIP30

   Run report interpretively C   QUIPREP

   Report to be :-
      Run in job or interactively(J/I)........[ ]
      Output to :-
        Terminal(T), System Printer(P)
        Disc(D), Attached Printer(A)..........[ ]

   Query XEQ file to be generated :-
      Name of Query XEQ file.................[       ]



----------------------------------------------------------------------------------
-
                  PARAMETERS FOR DISC OUTPUT

     Name of disc file.................................[       ]

     Suppress Headings and Titles.....................[N]

     Number of Lines per page.........................[66]
```

Menu	f2	f3	f4	f5	f6	f7	Exit

FIG. 2-13 Disc Output Parameters

Field Descriptions

Name of Disc file	This specifies the name of the disc file that will be generated by QUIP.
Number of Lines	This specifies the number of lines or rows that are printed between page breaks when Headings and Titles are not suppressed (maximum is 66, minimum is 22).

Job Passwords Screen

This screen is used to enter any necessary passwords. It is only used when a report is run in a job. QUIP will create a job stream file, whose name will be in the form QUIPIx, where x is a number in the range 1 to 9. If the files QUIPI1 through QUIPI9 already exist QUIP will then ask if one of the files can be deleted.

```
MAR 2 1987                                                  Ref:QUIP30

   Run report interpretively ℂ   QUIPREP

   Report to be :-
      Run in job or interactively(J/I)........[ ]
      Output to :-
        Terminal(T), System Printer(P)
        Disc(D), Attached Printer(A)..........[ ]

   Query XEQ file to be generated :-
      Name of Query XEQ file..................[       ]




   ------------------------------------------------------------------------------
   -

       Password for user     ℂ  JOHN          [         ]

       Password for group    ℂ  SMITH         [         ]

       Password for account  ℂ  LEDGERS       [         ]

   ┌──────────┬──────────┬──────────┬──────────┐ ┌──────────┬──────────┬──────────┬──────────┐
   │ Menu     │    f2    │    f3    │    f4    │ │    f5    │    f6    │   f 7    │   Exit   │
   └──────────┴──────────┴──────────┴──────────┘ └──────────┴──────────┴──────────┴──────────┘
```

FIG. 2-15 Job Passwords Screen

Field Descriptions

Password for user   Enter your user log-on password.

Password for group   Enter the log-on password for your local group.

Password for account   Enter the log-on password for your local account.

Note that after running the report interpretively the QUERY XEQ files will remain. These can be run again directly in QUERY, amended as required or submitted to the Q-GEN Compiler.

Note also that whether run interpretively or in compiled mode, when Terminal Output is selected the QUIP program switches to character mode before displaying the output on the terminal.

OPTION 3 COMPILING A REPORT

Compile Parameters Screen

This screen is used to enter the parameters to produce a report in compiled form.

```
MAR 2 1987                                              Ref:QUIP30

   Compile report   C  QUIPREP

   Report to be :-
      Run in job or interactively(J/I)........[ ]
      Output to :-
        Printer(P), Terminal(T) or Disc(D)....[ ]

   Cobol files to be generated :-
      Name of Object program..................[       ]
      Name of Source program(optional)........[       ]




   -------------------------------------------------------------------------------------
   -
```

Menu	f2	f3	Report Descr.		f6	f7	Exit

FIG. 2-16 Compile Parameters

Field Descriptions

Run in job or interactively	This will select the mode that the report will run in. If job (J) is selected the report will be scheduled as a job. If the report is selected to run interactively it will be run from the terminal. Any selection made by the user may be overridden by the system manager.
Output to	This controls whether the report will go to the printer, terminal or disc file. Output to printer will be 132 characters, Terminal 79 characters and Disc 128.
Name of Object program	This is the name of the generated object program. When compiled this file can be run directly using Option 4 on the main selection menu or by using the MPE RUN command.
Name of source	This is the name of the generated source file. This is an optional field. This option is used to produce a COBOL source file that can be amended to perform processing outside the scope of QUIP.
Note:	When compiling a report QUIP generates Query type XEQ statements which are passed to the Q-GEN compiler. These statements are written to a temporary file QUIPCODE which can be saved and edited as required. If the report is compiled in a job then the user will be required to enter the log-on passwords.

OPTION 4 RUN COMPILED REPORT

Compiled Report Screen

This screen is used to enter the parameter for running a report in compiled form. A compiled report can be produced using the previous option.

```
MAR 2 1987                                                    Ref:QUIP38

   Run report    C   QUIPROG

    Report to be:-
       Run in job or interactively(J/I)........[I]




    ------------------------------------------------------------------------------
    -
```

| Menu | f2 | f3 | f4 | f5 | f6 | f 7 | Exit |

FIG. 2-17 Run Compiled Report

Field Descriptions

Run in job or interactively   This will select the mode that the report will run in. If job (J) is selected the report will be scheduled as a job. If the report is selected to run interactively it will be run from the terminal.

Note: If the report is run as a Job then you will be prompted for the log-on passwords.

OPTION 6 MODIFY REPORT

The modify report option is used to amend a QUIP report. The modify option uses the same screens as the ADD report option. Screens are edited using the editing keys (insert,delete char) or by simply overwriting the displayed fields. This way it is possible to add and delete items to be reported, sorted etc.

The modifications are saved after enter has been pressed on the Report Title screen. The modification can be abandoned at any time before this by pressing the f1 (menu) function key.

OPTION 7 TEST REPORT

Test Parameters Screen

This option is used to preview the format of a QUIP report. This is done by selecting a subset of the available data. The subset is defined by the user, who specifies a limit on the number of records that should be selected. The screen shown below is used to enter the report test parameters.

```
MAR 2 1987                                         Ref:QUIP30

   Run report interpretively C  QUIPREP
    Number of records to be selected..........[10   ]
   Report to be :-
      Run in job or interactively(J/I)........[ ]
      Output to :-
        Terminal(T), System Printer(P)
        Disc(D), Attached Printer(A)..........[ ]

   Query XEQ file to be generated :-
      Name of Query XEQ file..................[       ]




   --------------------------------------------------------------------------------
   -
```

Menu		f3	Report Descr.	f5	f6	f7	Exit

FIG. 2-18 Interpretive Test Run Parameters

Field Descriptions

Num. of records   This specifies the number of records that should be selected.

Run in job or interactively   This will select the mode that the report will run in. If job (J) is selected the report will be scheduled as a job. If the report is selected to run interactively, it will run from the terminal. Any selection made by the user may be overridden by the system administrator.

Output to   This controls where the report will go. Output to a printer will be 132 characters wide. Output to the terminal will be 79 characters wide. Output to disc is 128 characters. The width of an attached printer is specified on a following screen.

Name of Query XEQ file   This is the name of the QUERY report file that will be generated. QUIP will actually generate 3 files: a database definition file (this will be suffixed with a D); a find definition file (suffixed with F); and finally a report definition file (suffixed with R).

SECTION 23 ₡ IMAGE TRENDS ANALYSIS

Introduction

FLEXIBASE/IMAGE Trends provides a set of facilities for the monitoring of structural trends within the IMAGE databases on HP3000 computer systems. FLEXIBASE/IMAGE Trends is easy to use, requiring little typing as it is Windows-based and presents information in a structured and easy to read manner. FLEXIBASE/IMAGE Trends can draw graphs and bar charts in windows, or present the statistics in numeric tabular form. The areas of information which can be monitored include Space Usage, Numbers of transactions and access efficiency.

Software Design

FLEXIBASE/IMAGE Trends has been designed for ease of use and employs Microsoft Windows as the user interface on Personal Computers. Each window has its own online help facility which explains the current window and the facilities available from it. The software operates in client/server mode reading statistical information stored in a statistics database on the HP3000. The statistics database on the HP3000 is updated by running the DIAGNOSE and ANALYZE modules of FLEXIBASE/IMAGE.

Software and Hardware Environments

FLEXIBASE/IMAGE Trends is designed for use with the IMAGE database system on the Hewlett Packard 3000 range of computers running the MPE/iX (Version 3.0 or later) operating system. Also required is an IBM PC (386SX or above) or compatible with a minimum 4MB of memory, running Microsoft Windows 3.1 or later, with Serial or a LAN connection to the Hewlett Packard computer.

General Principles of Operation

Data Collection

Before using the PC software to produce graphs of database performance, it is first necessary to collect data for analysis. The DIAGNOSE and ANALYZE modules store the data output in their reports in a database called FXSTAT.PUB.HQ (see the end of this section for a detailed description of the database); it is therefore necessary to allow write access to the database from the accounts where these modules are run (by default, the database is RELEASEd). The TRENDS function requires that a minimum of two collection points are present in the database before it can work.

The data is collected automatically. However, if it cannot be stored, either because the statistics database is inaccessible or because it has filled up, then no error message is given. It is up to the user to ensure that the database is properly maintained, and eventually to clear down or archive unwanted data.

In order to ensure that data is collected at meaningful, regular intervals, a parameter is maintained by FXUTIL which determines how often DIAGNOSE and ANALYZE will carry out data collection (see Section 24 on how to configure this parameter). By default, this parameter is set at 30 days. This means that, if you run DIAGNOSE (for example) on 1st September for the first time, data for 1st September will be loaded into the statistics database. However, if you then run DIAGNOSE during the month, the database will not be updated until 1st October. The same principle holds true for ANALYZE.

You should also note that, if you do not run ANALYZE, the corresponding data will of course not appear in your PC reports and graphs, although there is no indication of an "absence of data" in the PC product.

Running the PC Software

The application window is a sizable MDI-parent (Multiple Document Interface) window:

Menu and StatusBar

The functions of the menu items are explained under MENU. At the bottom of the window is a StatusBar. Only the first segment is currently used showing the Connection name when a connection is open.

Connections

Before any trending work can be commenced, a connection to a host computer must be opened. Connection facilities are available under Connection in the menu. Several connections may be open at the same time, each conversing with the statistics database on that particular HP3000, and it is possible to switch easily between connections. Thus it is simple to interrogate multiple host computers within a single run of FLEXIBASE/IMAGE Trends.

Data Services

The following data services are general and will operate on the data in whichever window is the active window. Using a combination of Searching, Subsetting, Sorting and Selected printing, there is a great deal of flexibility in building reports from statistical data in grids and graphs.

Print

The Print Control window is activated by selecting "Print" under "File" in the menu. It allows specification of the data to be printed. The Window field at the top displays the active window caption (the Print function works with respect to the currently active window). Usually there is only one set of data in a window which is printable but where there are more, then the data dropdown list allows selection of which data is required. If the selected data is in a grid, then the grid columns are individually multi-selectable for reporting - and note that the printed width of any column will be as the width is currently set in the grid. If the data is a non-grid list or text, then there is a choice to print selected data or all data. The current printer font settings are shown and can be changed by pressing the "Change Font" button to get access to a standard Font definition dialog box. (Access to the standard Printer setup dialog box is available under "File" in the menu.) For a grid the range of data to be printed is flexible by reference to the beginning, end, selected row and required number of rows. Pressing the "OK" button actions the print.

Save Data to PC File & Extended Copy to Clipboard

The "Save Data" window is activated by selecting "Save Data" under "File" in the menu. It allows specification of the data to be saved. The Extended Copy to Clipboard is activated by selecting "Extended Copy to Clipboard" under "Edit" in the menu.

The "Extended Copy to Clipboard" enables detailed specification of the data to be copied, unlike the standard copy to clipboard which simply copies selected data. Its specification window is exactly the same as for the save specification except that the spreadsheet option is not available.

The Window field at the top displays the active window caption (the "Save Data" function works with respect to the currently active window). Usually there is only one set of data in a window which can be saved but where there are more, then the data dropdown list allows selection of which data is required. If the selected data is in a grid, then the grid columns are individually multi-selectable for saving - and note that the saved width of any column will be as the width is currently set in the grid. If the data is a non-grid list or text, then there is a choice to save selected data or all data. A checkbox option is available to save the data in spreadsheet format (i.e. in comma-separated format) which can be imported into spreadsheet applications.

For a grid the range of rows to be saved is flexible by reference to the beginning, end, selected row and required number of rows. The data will be saved in ASCII format unless the data is a graph in which case it will be saved in bitmap format. Pressing the "OK" button causes a dialog box to appear requesting the filename for the new data file.

Find

The Find or Search window is activated by selecting "Find" under "Edit" in the menu. The Window field at the top displays the active window caption (the Find function works with respect to the currently active window). Usually there is only one set of data in a window which is searchable but where there are more, then the data dropdown list allows selection of which data is to be searched. If the selected data is in a grid, then an individual grid column must be selected for searching. Key into the "Search for" field the text which is to be searched for, and check the Case sensitive checkbox if it is appropriate, else the search will be case insensitive. If the "Report each find" radio-button is checked, (this is the only method for text searches) then pressing the "Find" button initiates a search from the beginning or from the latest find. When a find is made then the found text is highlighted in the active window and a beep sounds. (It is therefore a good idea to position the Find window so that you can still see the relevant contents of the active window.) Pressing the "Find" button again then looks for the next occurrence of the search text, and so on. An alternative method of search, for grids only, is selected by clicking one of the radio-buttons relating to checkmarks. In these cases, pressing the "Find" button searches the whole grid and and inserts a checkmark at the left end of every row where a find is present. Previous checkmarks are optionally retained. A further function is then available. By pressing the "Subset" button, all un-CheckMarked rows are removed from the grid. Note that there are two related functions available under "Edit" in the menu - "Delete All CheckMarks" and "Reverse All CheckMarks". Using these facilities in combination, it is possible to manipulate the grid contents to form compound subsets.

Pattern Matching in the Search For Field

Built-in pattern matching provides a versatile tool for string comparisons. The pattern-matching features allow you to use wildcard characters, such as those recognized by the operating system, to match strings. The wildcard characters and what they match are shown in the following table:

?	Any single character
*	Zero or more characters
#	Any single digit (0-9)
[charlist]	Any single character in charlist
[!charlist]	Any single character not in charlist

A group of one or more characters (charlist) enclosed in brackets ([ ]) can be used to match any single character in expression and can include almost any characters in the ANSI character set, including digits. In fact, the special characters left bracket ([ ), question mark (?), number sign (#), and asterisk (*) can be used to match themselves directly only by enclosing them in brackets. The right bracket ( ]) cannot be used within a group to match itself, but it can be used outside a group as an individual character. In addition to a simple list of characters enclosed in brackets, charlist can specify a range of characters by using a hyphen (-) to separate the upper and lower bounds of the range. For example, [A-Z] in pattern results in a match if the corresponding character position in expression contains any of the uppercase letters in the range A through Z. Multiple ranges are included within the brackets without any delimiting. For example, [a-zA-Z0-9] matches any alphanumeric character.

Other important rules for pattern matching include the following:

An exclamation point (!) at the beginning of charlist means that a match is made if any character except the ones in charlist are found in expression. When used outside brackets, the exclamation point matches itself.

The hyphen (-) can appear either at the beginning (after an exclamation mark if one is used) or at the end of charlist to match itself. In any other location, the hyphen is used to identify a range of ANSI characters.

When a range of characters is specified, they must appear in ascending sort order (from lowest to highest). [A-Z] is a valid pattern, but [Z-A] is not.

The character sequence [ ] is ignored; it is considered to be a zero-length string.

In some languages, there are special characters in the alphabet that actually represent two separate characters. For example, several languages use the character "æ" to represent the characters "a" and "e" when they appear together. The Like operator recognizes that the single special character and the two individual characters are equivalent. When a language that uses one of these special characters is specified in the WIN.INI file (sLanguage), an occurrence of the special single character in either pattern or expression matches the equivalent 2-character sequence in the other string. Similarly, a special single character in pattern enclosed in brackets (by itself, in a list, or in a range), matches the equivalent 2-character sequence in expression.

Sort

The sort window is activated by selecting "Sort" under "Edit" in the menu. The Window field at the top displays the active window caption (the Sort function works with respect to the currently active window). Usually there is only one set of data in a window which is sortable but where there are more, then the data dropdown list allows selection of which data is to be sorted. Only data in grids can be sorted. Grids can be sorted on up to three columns as keys with options for Ascending/Descending order and Case-sensitivity on each key. Click on each column required as a key, primary key first. A "Reset" keys button is available to enable re-selection of keys.

Press the "Sort" button to action the Sort. The grid data is re-arranged in situ.

Connections to Host Computer

FLEXIBASE/IMAGE Trends is designed to run multiple simultaneous connections which can be switched between, and these connections may be any mixture of Reflection, Session or Serial. It would also be possible to run multiple copies of FLEXIBASE/IMAGE Trends each running various connections. A connection is specified using the Connection configuration window which is displayed by selecting "New" or "Modify" under "Connections" in the menu. If New was selected then the fields will all contain default values, else if Modify was selected the fields contain the parameters of the existing selected connection. Using Modify, if the connection name is changed, then when the "Save" button is pressed a new connection file is created under the new name, whereas if the connection name is unchanged then the existing connection file will be modified. Depending upon the Operating system and Connection Type chosen, the appropriate fields will be made visible.

Detailed explanations of how to configure connections using the PC's Serial Port, Reflection, Business Sessions or HPAdvanceLink for Windows, are provided in the MS-Write files which are installed from diskette at the same time as the PC software.

Opening a Connection

Clicking on Open under Connections in the Menu-bar causes a dialog box to appear. This dialog box enables selection of a configuration file from those listed. Configuration files have the file extension ".fxc" and must reside in the installation subdirectory. Three example template configuration files are supplied with this software named "serial.fxc" and "sessionx.fxc" and "reflecx.fxc". These files are intended as examples rather than defaults, and the user is urged to modify them to produce appropriate connection configurations before attempting to Open a connection (see Modify Connection and New Connection).

The opening process goes through the following stages:

- Establishes connection with (response from) the host.
- Logs on, as configured in the connection file - passwords will be requested if not configured.
- Runs the host server software.
- Opens the FXSTAT statistics database.

The process is, however, subject to modification by checking and un-checking of options in the Open dialog box - namely a Terminal Window can be presented to the user before connection (this enables the user to route through data switches, check response from host or to effect a non-standard logon etc.) and the Logon can be omitted (in order to maintain an existing logon, or where the user is logging on with the Terminal window).

When a connection is Opened which is configured to use a Terminal emulator, FLEXIBASE first looks for a running copy of Session or Reflection which is appropriately configured (see the next section). If one is then FLEXIBASE forms a DDE link to that copy of Session or Reflection for its communication. If a suitable copy is not already running then a copy of Session or Reflection is caused to run automatically and the link then formed.

Running FLEXIBASE/IMAGE with a TERMINAL EMULATOR

Users who can only connect to the host by a non-serial (i.e. network) connection will need to use this method, which employs a DDE link between the terminal emulator and FLEXIBASE. At present this method is only supported with REFLECTION from Walker Richer & Quinn (Version 4.0 and upwards), SESSION from Unison/Tymlabs or ADVANCELINK from Hewlett Packard (Version 3.18 and upwards). The applications must be running under Windows 3.1 or later.

There is an exclusive relationship between a configured copy of the terminal emulator, and a FLEXIBASE connection file, based on the DDE Application Server Name in the configuration of the Terminal Emulator. It is therefore possible for a FLEXIBASE connection to link to its configured partner emulator if that is already running, or otherwise to run and link to a new copy.

Trend Analysis

This function is activated by selecting "Trends" in the Menu. Using this function, the DBA may graphically analyze various parameters relating to data and filespace size, access efficiency and transaction volumes by time. Various graph and chart styles are available, and time periods and individual parameters of individual objects are selectable and can be combined on the same graph.

Specification of a Graph or Chart

Objects and parameters

There are four types of Object for which Trend Data is available - DBs, DataSets, Paths, Programs - represented by the four List Objects:- radio-buttons at the top left of the window. In addition there is a radio-button for an Hourly analysis which enables the transactions volumes during the specified period to be analyzed by the 24 hours of the day. Turning on one of these radio-buttons causes a grid of available object names to appear beneath the radio-buttons.

The first time that an object-type is selected, the list will be obtained from the server. Subsequently selecting that object-type simply makes the list visible. Any time that a Database name is selected (clicked-on), it becomes the Current Database, and its name is shown in the current database field at the top right of the window. For Sets, Paths and Programs, the list will be limited to those objects relevant to the current database. The choice of object type also determines which parameters are available and displayed in the parameters listbox on the right of the window beneath the current database field. At the bottom of the window is the Selected Trend Data grid. When the Add button is pressed (or a double-click made in the Object or Parameter lists), the currently selected object and parameter are added into this grid. Each row in this grid will give rise to a line on the graph. The number of lines on the graph is limited only by clarity, and the user may mix different object-types and parameters on the same graph, the scale for each parameter being automatically rationalized to bring the lines into the same visual range. Entries may be removed from this grid by multi-selecting them and pressing the Remove button.

It is important to remember that some statistics are derived from DIAGNOSE runs and some from ANALYZE runs. Therefore if ANALYZE is never run, only certain statistics will be available. For a full listing of the FXSTAT database schema, and a description of how FLEXIBASE loads data into it, see section 3.8.

The following parameters are available for plotting (see below for a detailed description of each parameter):-

Databases
Average DBPuts per day
Average DBDeletes per day
Average DBUpdates per day
Average DBOpens per day
Average DBCloses per day
Average Aborts per day
Average Abnormal Ends per day
Maximum Concurrent Users
Maximum DBPuts in one day
Maximum DBDeletes in one day
Maximum DBUpdates in one day

DataSets
Average DBPuts per day
Average DBDeletes per day
Average DBUpdates per day
Average NullDBUpdates per day
Capacity
Number of Entries
Maximum Increase in number of entries in one day
Size of set in sectors
Used space in sectors
% Messy (detail sets)
High Water Mark (detail sets)
% Deleted
% Full
Average Secondary Chain length (master sets)
% Inefficient (master sets)
% Secondaries (master sets)
Clustering factor (master sets)

Paths
The format of the Path Name is:

MasterSetName . DetailSetName (SearchItemName)

Average Chain Length
Maximum Chain Length
Minimum Chain Length

Programs
Average DBPuts per day
Average DBDeletes per day
Average DBUpdates per day
Average NullDBUpdates per day

24-Hourly Analysis
Average DBPuts per hour
Average DBDeletes per hour
Average DBUpdates per hour
Average DBOpens per hour
Average DBCloses per hour
Average Aborts per hour
Average Abnormal Ends per hour

Dates
The user can overtype the default dates in the From and To fields in order to limit the data plotted
to a particular date range. Numeric dates follow the format entered under International in the
Windows Control Panel. The default dates allow all data to be plotted. The user need not be

concerned whether objects within the same graph have their dates of trend data collection synchronized (e.g. if an object is new and has less trend data collected) since the software will always produce visually correct graphics. There must be at least two events of trend data collection from DIAGNOSE and/or ANALYZE before a graph can be drawn. The dates (and times) held in the FXSTAT.PUB.HQ database on the HP3000 relate to the start times of DIAGNOSE and Analyze runs.

Graph Type
The Graph Definition area allows a title for the graph to be overtyped into the Title field, and various options to be selected for the type of graph using the radio-buttons. In addition to Line Graphs which are time-based i.e. the X-axis accurately represents the dates of data collection, the user can also choose Bar Charts which represent each event of trend data collection with a bar i.e. the X-axis does not accurately correspond to time, but if DIAGNOSE and/or ANALYZE has been run on a reasonably regular basis, then the X-axis will be satisfactory.

There are several styles of Bar Chart available, and, with the Horizontal bars option, a Dates checkbox allows the actual dates of trend data collection to displayed against each bar.

Loading and Saving Graph Specifications
The specification of a graph may be saved to a file on disc by selecting Save Specification under File in the Menu, and loaded back by selecting Open Specification under File in the Menu. (Trend Graph Specification files are saved with the extension .TGR.)

Activating the Graph or Bar Chart
Once the graph has been fully specified, it can be displayed in a window by pressing the Draw Graph button. Alternatively, pressing the Data to Grid button will cause the specified trend data to be displayed in a grid, in which case the specification of Graph Type is not used, only the object parameters and dates are relevant. Graphs can be printed by selecting Print under File in the Menu.

Menu Options

File
Open Specification...   Causes a dialog box to be displayed requesting the name of the file containing the trend graph specification to be opened.

Save Specification...   Causes a dialog box to be displayed requesting the name of the file under which to save the trend graph specification.

Save Data...   Causes a window to be displayed for specifying the details of the data from the active window to be saved to a PC file.

Print...   Causes a window to be displayed for specifying the details of the data from the active window to be printed.

Printer Setup...   Causes a standard printer setup dialog box to be displayed.

Exit   Causes Termination of the application after a confirmation. A subsequent dialog box will ask how each open connection is to be terminated.

Edit
Copy to Clipboard   Causes selected text in the active window to be copied to the clipboard, (or use Ctrl/C).

Extended Copy to Clipboard   Causes a window to be displayed for specifying the details of the data from the active window to be copied to the clipboard.

Paste   Causes text on the clipboard to be pasted into the selected text of the active window.

Find... | Causes a window to be displayed for specifying the details of the data to be searched for in the active window.

Delete All Checkmarks... | Causes all Checkmarks in the active window grid to be deleted.

Reverse All Checkmarks... | Causes all Checkmarks in the active window grid to be deleted, whilst a checkmark appears in all grid rows without one.

Susbset CheckMarked rows | Causes all rows in the active window grid to be deleted unless they contain a Checkmark (following a count and confirmation).

Sort... | Causes a window to be displayed for specifying the details of the data to be sorted in the active window.

Connections
Open | Causes a dialog box to be displayed listing connection files so that one can be selected for opening.

Close | Closes the currently open connection. A dialog box asks whether a log off is to be effected.

Switch | Causes a dialog box to be displayed listing the currently opened connections so that one can be selected for switching to.

New | Causes a window to be displayed for specifying the details of a new connection.

Modify | Causes a dialog box to be displayed listing connection files so that one can be selected for modifying.

Delete | Causes a dialog box to be displayed listing connection files so that one can be selected for deleting.

Trends
Causes a window to be displayed allowing specification of the trend data required.

Windows
Cascade | Causes all MDI-child windows to be displayed in cascaded fashion.

Tile Vertical | Causes all MDI-child windows to be displayed in vertically tiled fashion.

Tile Horizontal | Causes all MDI-child windows to be displayed in horizontally tiled fashion.

Arrange Icons | Re-arranges all MDI-child window icons into order.

Close All Windows | Closes all MDI-child windows.

Window Caption List | All MDI-child windows, whether iconised or not, are listed here. Clicking on any listed caption brings that window to the front with focus.

Help
Contents | Causes Windows Help to display the alphabetical index of help topics for FLEXIBASE/SQL.

Active Window | Causes Windows Help to display the context sensitive help for the active window. The F1 key will also activate this so that help can still be obtained even when there is a modal dialog box on screen.

General Principles of operation | Causes Windows Help to display details of the General Principles of Operating this software.

Overview   Causes Windows Help to display an Overview of the software funtionality.

About FLEXIBASE/IMAGE Trends   Causes information to be displayed on the version number of
FLEXIBASE/IMAGE Trends PC and Host software which is being used.

TRENDING PARAMETERS

The following paragraphs contain a detailed description of the definitions of Data Values Displayed
concerning Data Bases, Data Sets, Paths, Programs and 24-Hourly Analysis.

Databases

DateRun, TimeRun
Time of start of Diagnose or Analyse Run

Average DBPuts per day
The total number of DBPuts issued to all sets within this logfile within the specified database
divided by the number of days the logfile being added to the history file (a file called
"dbname"+FX containing historical records of full analyze runs) was running.

Average DBDeletes per day
The total number of DBDeletes issued to all sets within this logfile within the specified database
divided by the number of days the logfile being added to the history file (a file called
"dbname"+FX containing historical records of full analyze runs) was running.

Average DBUpdates per day
The total number of all DBUpdates issued to all sets within this logfile within the specified
database divided by the number of days the logfile being added to the history file (a file called
"dbname"+FX containing historical records of full analyze runs) was running.

Average DBOpens per day
The total number of DBOpens issued to all sets within this logfile within the specified database
divided by the number of days the logfile being added to the history file (a file called
"dbname"+FX containing historical records of full analyze runs) was running.

Average DBCloses per day
The total number of DBCloses issued to all sets within this logfile within the specified database
divided by the number of days the logfile being added to the history file (a file called
"dbname"+FX containing historical records of full analyze runs) was running.

Average Aborts per day
The total number of program aborts detected by the logging process within this logfile divided by
the number of days the logfile being added to the history file (a file called "dbname"+FX
containing historical records of full analyze runs) was running. A program abort is defined as a
process which ended without explicitly calling DBClose.

Average Abnormal Ends per day
The total number of Abnormal transaction ends caused by any Program on the specified database
within this logfile divided by the number of days the logfile being added to the history file (a file
called "dbname"+FX containing historical records of full analyze runs) was running. In this case
the program has issued a DBBegin to start a logical transaction but has failed to issue a DBEnd
before finishing.

Maximum Concurrent Users

This figure is the largest observed number of concurrently operating write users at any one time. This occurrence may not have been in the current logfile, but represents an historical maximum. This figure only represents write users, as transaction logging does not record read-only users (DBOpen modes 5-8). Thus, this figure is not a complete figure for the maximum number of users on the database concurrently. However, by multiplying this figure by a factor of, say, 1.5 an approximate value can be reached. This multiplying factor will, of course, be different for each database. It should be noted that programs which only read the database can still open the database in modes 1 to 4 and thus be logged.

Maximum DBPuts in one day

For each logfile the total number of puts is divided by the number of days for which the logfile ran. The maximum observed value of this figure is stored in the database Analyze History file (a file called "dbname"+FX). When compared with the average number of puts (calculated by dividing the total days into the total puts) the volatility of the put access to the database can be judged. A large maximum with a small average indicates an abnormally large addition run which is run infrequently. Comparable numbers indicate a smoother process of adding data entries.

Maximum DBDeletes in one day

For each logfile the total number of deletes is divided by the number of days for which the logfile ran. The maximum observed value of this figure is stored in the database's Analyze History file (a file called "dbname"+FX). When compared with the average number of deletes (calculated by dividing the total days into the total deletes) the volatility of the delete access to the database can be judged. A large maximum with a small average indicates an abnormally large deletion run which is run infrequently. Comparable numbers indicate a smoother process of deleting data entries.

Maximum DBUpdates in one day

For each logfile the total number of updates is divided by the number of days for which the logfile ran. The maximum observed value of this figure is stored in the database's Analyze History file (a file called "dbname"+FX). When compared with the average number of updates (calculated by dividing the total days into the total updates) the volatility of the update access to the database can be judged. A large maximum with a small average indicates an abnormally large update run which is run infrequently. Comparable numbers indicate a smoother process of updating data entries.

Data Sets

Average DBPuts per day

The total number of DBPuts issued to any set within this logfile within the specified database divided by the number of days the logfile being added to the history file (a file called "dbname"+FX containing historical records of full analyze runs) was running.

Average DBDeletes per day

The total number of DBDeletes issued to any set within this logfile within the specified database divided by the number of days the logfile being added to the history file (a file called "dbname"+FX containing historical records of full analyze runs) was running.

Average DBUpdates per day

The total number of all DBUpdates issued to any set within this logfile within the specified database divided by the number of days the logfile being added to the history file (a file called "dbname"+FX containing historical records of full analyze runs) was running.

Average NullDBUpdates per day

The total number of DBUpdates issued by any Program to any set within this logfile within the specified database which did not actually change any data item values, divided by the number of days the database's logfile being added to the history file (a file called "dbname"+FX containing historical records of full analyze runs) was running. This statistic is very important to program performance. It represents uneccessary database calls, and thus redundant I/O. This quantity is

also included in the Num Updates figure above, and thus comparison of these two figures gives an indication of the potential for performance improvement.

Capacity

The maximum number of possible entries in the set.

Number of Entries

The number of entries read by the program in the set, using a serial read.

Maximum Increase in number of entries in one day

When adding or deleting entries to a dataset, free space in the dataset is consumed or released. In a typical month there may well be an extreme program which performs a large number DBPUTs. The Capacity of a dataset must at least be sufficient to accept all of these new entries without filling up. Thus, ANALYZE computes the maximum increase in the number of entries for each set (as being the largest difference between the number of DBPUTs and DBDELETEs for each logfile). The maximum observed figure is stored and reported here. Thus, when estimating a Capacity it must be at least this many entries greater than the current entry count.

Size of set in sectors

The total disc space taken up by the dataset expressed in sectors (4096 Sectors = 1Mb). IMAGE builds the dataset to the size specified by the capacity, even if no entries are added to the set. This value represents the total amount of disc space consumed by the dataset.

Used space in sectors

The approximate disc space used by the actual entries in the dataset expressed in Sectors. This value is calculated by multiplying the total disc space ("Set Size" above) by the number of entries and dividing by the Capacity. In fact there could be "dead" space in the set at the end of each block but this is not accounted for in the calculation. This figure subtracted from the total disc space gives a rough estimate of the amount of "wasted" disc space taken up by the dataset.

% Messy (detail sets)

This is the percentage of non-contiguous chained entries along the Primary Path pointing out of the block (i.e. entries in the chain do not point to the next sequential entry in the set and point out of the current block). A perfectly REPACKed detail set would have zero Percentage Messy. The higher this figure is, the longer and more inefficient chained access along the Primary Path becomes. This Percentage Messy is measured using a "block" size of 96 Sectors instead of the set's block size.

High Water Mark (detail sets)

The Maximum Number of Entries that the set has ever contained. For detail sets this will be equal to the number of entries plus the number of deleted entries. This represents the largest number of entries which this set has ever contained, since the last Repack. A Repack will reduce this number to equal the number of entries.

% Deleted

For a Detail Set, this is the Number of Deleted Entries as a percentage of the Maximum Number of Entries that the set has ever contained since the last, if any, REPACK (i.e., the "Highwater Mark").

% Full

The number of entries read as a percentage of the Capacity.

Average Secondary Chain length (master sets)

The average length of the Secondary Chains in a Master Set. A Primary with no Secondaries is considered to have a chain length of one.

% Inefficient (master sets)

For a Master Set, this is the number of Secondaries in the set located in a different block to that of the Primary, as a percentage of the number of entries in the set. This is the prime indicator of inefficiency in the set as it causes extra disc accesses or memory transfers to read the entries.

% Secondaries (master sets)

The number of Secondary entries in a Master Set as a percentage of the number of entries in the set. A high percentage can result in poor performance if associated with a high Percentage Inefficient (see next item).

Clustering factor (master sets)

An indicator of how much the entries in a Master Set cluster together. A high value can indicate poor performance, especially if associated with a high percentage of Secondaries. Range is 0.0 to 1.0, with values over 0.25 considered as high.

Paths

The format of the Path Name is MasterSetName + '.' + DetailSetName + '(' + SearchItemName + ')', e.g. MASTER.DETAIL(VALUE).

Average Chain Length

Each "Path" is a link between a MasterSet and a DetailSet by means of a specific SearchItem. Each entry in the Master Set may have zero or more entries in the Detail Set with its' specific Search Item value. The number of entries for this SearchItem value is termed the "Chain Length". This value is the average of all Chain Lengths on this path.

Maximum Chain Length

Each "Path" is a link between a MasterSet and a DetailSet by means of a specific SearchItem. Each entry in the Master Set may have zero or more entries in the Detail Set with its' specific Search Item value. The number of entries for this SearchItem value is termed the "Chain Length". This value is the maximum observed Chain Length on this path.

Minimum Chain Length

Each "Path" is a link between a MasterSet and a DetailSet by means of a specific SearchItem. Each entry in the Master Set may have zero or more entries in the Detail Set with its' specific Search Item value. The number of entries for this SearchItem value is termed the "Chain Length". This value is the minimum observed Chain Length on this path.

Programs

Average DBPuts per day

The total number of DBPuts issued by this Program to any set within this logfile within the specified database divided by the number of days the logfile being added to the history file (a file called "dbname"+FX containing historical records of full analyze runs) was running.

Average DBDeletes per day

The total number of DBDeletes issued by this Program to any set within this logfile within the specified database divided by the number of days the logfile being added to the history file (a file called "dbname"+FX containing historical records of full analyze runs) was running.

Average DBUpdates per day

The total number of all DBUpdates issued by this Program to any set within this logfile within the specified database divided by the number of days the logfile being added to the history file (a file called "dbname"+FX containing historical records of full analyze runs) was running.

Average NullDBUpdates per day

The total number of DBUpdates issued by this Program to any set within this logfile within the specified database which did not actually change any data item values, divided by the number of

days the logfile being added to the history file (a file called "dbname"+FX containing historical records of full analyze runs) was running. This statistic is very important to program performance. It represents uneccessary database calls, and thus redundant I/O. This quantity is also included in the Num Updates figure above, and thus comparison of these two figures gives an indication of the potential for performance improvement.

24-Hourly Analysis

Average DBPuts per hour
The total number of DBPuts issued within this hour of the day to any set within the specified database divided by the number of days between the start and end of the graph readings.

Average DBDeletes per hour
The total number of DBDeletes issued within this hour of the day to any set within the specified database divided by the number of days between the start and end of the graph readings.

Average DBUpdates per hour
The total number of all DBUpdates issued within this hour of the day to any set within the specified database divided by the number of days between the start and end of the graph readings.

Average DBOpens per hour
The total number of DBOpens issued within this hour of the day to any set within the specified database divided by the number of days between the start and end of the graph readings.

Average DBCloses per hour
The total number of DBCloses issued within this hour of the day to any set within the specified database divided by the number of days between the start and end of the graph readings.

Average Aborts per hour
The total number of program aborts detected by the logging process within this hour of the day divided by the number of days between the start and end of the graph readings. A program abort is defined as a process which ended without explicitly calling DBClose. Ideally this figure should be zero, but could be one if the program terminated abnormally. However, it could also be one if the program simply failed to call DBClose before ending.

Average Abnormal Ends per hour
The total number of Abnormal transaction ends caused by any Program within this hour of the day in the specified database divided by the number of days between the start and end of the graph readings. In this case the program has issued a DBBegin to start a logical transaction but has failed to issue a DBEnd before finishing. Ideally this figure should be zero, but if the program aborted the log process would automatically post Abend entries signifying an abnormal end.

Trend History Database

As of release E.04.01 both Diagnose and Analyze will store results of their runs in a Trend History Database called FXSTAT.PUB.HQ. The information stored in this database can be accessed by the user using QUERY.PUB.SYS or by using the FLEXIBASE/IMAGE Trends Windows PC program. Data will be stored in this database whenever a full Diagnose or Analyze (Stats Option - updating the History File, see section 5) run is performed. The user does not need to run both of these programs in order to use the Trends Database, however for a complete picture it is recommended that both runs are done regularly.

Data will only be stored from a Diagnose run if the user selects to Diagnose all datasets in the database (i.e. replying "@" in response to the dataset name). In addition the FXUTIL utility allows the user to configure the minimum duration (in number of days) between Diagnose runs on the same database that data is allowed to be stored. The default is 30 days, and thus Diagnose will only store data in the FXSTAT.PUB.HQ database if 30 days or more have elasped since the last time data was stored.

Data will only be stored from an Analyze run if the user selects the Stats option, is the creator of the database, and selects Yes for the question "OK to Update History File". In addition the FXUTIL utility allows the user to configure the minimum duration (in number of days) between Analyze runs on the same database that data is allowed to be stored. The default is 30 days, and thus Analyze will only store data in the FXSTAT.PUB.HQ database if 30 days or more have elasped since the last time data was stored.

The database schema for the FXSTAT.PUB.HQ database is as follows:

```
$CONTROL BLOCKMAX=2048
Begin Data Base FXSTAT ;
Passwords:
          5  ISB       ;
         10  ROBIN     ;
Items:
<<     1 >>   ABNORMAL-ENDS,    24 I  2  (5/10);
<<     2 >>   ABORTS,           24 I  2  (5/10);
<<     3 >>   ACTIONED,            X  2  (5/10);
<<     4 >>   AVERAGE-LENGTH,      R  4  (5/10);
<<     5 >>   AVGE-SEC-CHAIN,      R  4  (5/10);
<<UN-REF>>    BITMAP-LENGTH,       I  1  (5/10);
<<     7 >>   BLOCKING-FACTOR,     I  1  (5/10);
<<UN-REF>>    BLOCK-LENGTH,        I  1  (5/10);
<<     9 >>   BLOCKMAX,            I  1  (5/10);
<<    10 >>   CAPACITY,            I  2  (5/10);
<<    11 >>   CHAIN-LENGTH,        I  2  (5/10);
<<    12 >>   CHAIN-NUMBER,        I  2  (5/10);
<<    13 >>   CLOSES,           24 I  2  (5/10);
<<    14 >>   CLUSTERINGFACTOR,    R  2  (5/10);
<<    15 >>   CONTROL-CODE,        X  2  (5/10);
<<UN-REF>>    DATE-CREATED,        X  8  (5/10);
<<UN-REF>>    DATE-DBSTORED,       X  8  (5/10);
<<    18 >>   DATE-RUN,            X  8  (5/10);
<<    19 >>   DELETES,             I  2  (5/10);
<<    20 >>   DETAIL-SETNAME,      X 16  (5/10);
<<    21 >>   DATABASE-NAME,       X 26  (5/10);
<<    22 >>   EXCLUSIVE,           X  2  (5/10);
<<    23 >>   ENTRY-LENGTH,        I  1  (5/10);
<<    24 >>   FORMATVERSION,       X  8  (5/10);
<<    25 >>   GETS,                I  2  (5/10);
<<    26 >>   GLOBAL-MISCVAL1,     I  2  (5/10);
<<    27 >>   GLOBAL-MISCVAL2,     I  2  (5/10);
<<    28 >>   GLOBAL-MISCVAL3,     I  2  (5/10);
<<    29 >>   GLOBAL-MISCVAL4,     R  4  (5/10);
<<    30 >>   GLOBAL-MISCVAL5,     R  4  (5/10);
<<UN-REF>>    GLOBAL-MISCVAL6,     R  4  (5/10);
<<    32 >>   GLOBAL-MISCVAL7,     X 10  (5/10);
<<    33 >>   GLOBAL-MISCVAL8,     X 10  (5/10);
<<UN-REF>>    GLOBAL-MISCVAL9,     X 10  (5/10);
<<    35 >>   HIGHWATERMARK,       I  2  (5/10);
<<UN-REF>>    HISTFILEPRESENT,     X  2  (5/10);
<<    37 >>   HOUR-MISCVAL1,    24 I  2  (5/10);
<<UN-REF>>    HOUR-MISCVAL2,    24 I  2  (5/10);
<<    39 >>   HOUR-MISCVAL3,    24 R  4  (5/10);
<<UN-REF>>    HOUR-MISCVAL4,    24 R  4  (5/10);
<<UN-REF>>    IMAGELEVEL,          X  2  (5/10);
<<UN-REF>>    IMAGEMINBUFFERS,     I  1  (5/10);
<<    43 >>   IMPLIED-UPDATES,     I  2  (5/10);
<<    44 >>   LDEV-NUM,            I  1  (5/10);
<<UN-REF>>    LOGID,               X  8  (5/10);
<<    46 >>   MASTER-SETNAME,      X 16  (5/10);
<<UN-REF>>    MAX-BUFFERS,         I  1  (5/10);
<<UN-REF>>    MAX-BUFFSPECS,       I  1  (5/10);
<<    49 >>   MAX-CONUSERS,        I  2  (5/10);
<<    50 >>   MAX-DELETEPERDAY,    R  2  (5/10);
<<    51 >>   MAX-FREESPACE,       R  4  (5/10);
<<    52 >>   MAX-GETSPERDAY,      R  2  (5/10);
<<    53 >>   MAX-INCREASE,        I  2  (5/10);
<<    54 >>   MAX-LENGTH,          I  2  (5/10);
<<    55 >>   MAX-LOGRECS,         I  2  (5/10);
<<    56 >>   MAX-PUTSPERDAY,      R  2  (5/10);
<<    57 >>   MAX-UPDATEPERDAY,    R  2  (5/10);
<<UN-REF>>    MEDIA-LENGTH,        I  1  (5/10);
<<    59 >>   MIN-LENGTH,          I  2  (5/10);
<<    60 >>   MIN-BUFFERS,         I  1  (5/10);
```

```
<<    61 >>    MIN-BUFFSREQ,       I  1  (5/10);
<<UN-REF>>     MPE-RECORD-SIZE,    I  1  (5/10);
<<    63 >>    NEWVALUE,           X 10  (5/10);
<<    64 >>    NULL-UPDATES,       I  2  (5/10);
<<    65 >>    NUM-ABORTS,         I  2  (5/10);
<<    66 >>    NUM-ABNORMALENDS,   I  2  (5/10);
<<UN-REF>>     NUM-BLOCKS,         I  2  (5/10);
<<    68 >>    NUM-CONCUSERS,      I  2  (5/10);
<<    69 >>    NUM-CONTBLOCKS,     I  2  (5/10);
<<    70 >>    NUM-CONTENTRIES,    I  2  (5/10);
<<    71 >>    NUM-CLOSES,         I  2  (5/10);
<<    72 >>    NUM-DAYS,           R  2  (5/10);
<<    73 >>    NUM-DAYSHIST,       R  2  (5/10);
<<UN-REF>>     NUM-DBINFOENTRY,    I  2  (5/10);
<<    75 >>    NUM-DELETED,        I  2  (5/10);
<<    76 >>    NUM-DELETES,     24 I  2  (5/10);
<<    77 >>    NUM-ENTRIES,        I  2  (5/10);
<<    78 >>    NUM-GETS,           I  2  (5/10);
<<    79 >>    NUM-INDEXFILES,     I  1  (5/10);
<<    80 >>    NUM-INDEXSETS,      I  1  (5/10);
<<UN-REF>>     NUM-INEFPOINTS,     I  2  (5/10);
<<    82 >>    NUM-ITEMS,          I  1  (5/10);
<<    83 >>    NUM-OPENS,          I  2  (5/10);
<<    84 >>    NUM-PATHS,          I  1  (5/10);
<<    85 >>    NUM-PUTS,        24 I  2  (5/10);
<<    86 >>    NUM-SETS,           I  1  (5/10);
<<    87 >>    NUM-SECONDARIES,    I  2  (5/10);
<<    88 >>    NUM-SORTEDPATHS,    I  1  (5/10);
<<    89 >>    NUM-UPDATES,     24 I  2  (5/10);
<<    90 >>    NUM-WRITEACCESS,    R  4  (5/10);
<<    91 >>    OPENS,           24 I  2  (5/10);
<<    92 >>    OLDVALUE,           X 10  (5/10);
<<    93 >>    PATH-NUMBER,        K  1  (5/10);
<<    94 >>    PATH-MISCVAL1,      I  2  (5/10);
<<    95 >>    PATH-MISCVAL3,      R  4  (5/10);
<<    96 >>    PATH-MISCVAL5,      X 10  (5/10);
<<    97 >>    PERCENT-DELETED,    R  2  (5/10);
<<    98 >>    PERCENT-FULL,       R  2  (5/10);
<<    99 >>    PERCENT-INEFF,      R  2  (5/10);
<<   100 >>    PERCENT-MESSY,      R  2  (5/10);
<<   101 >>    PERCENT-MESSY2,     R  2  (5/10);
<<   102 >>    PERCENT-MESSY3,     R  2  (5/10);
<<   103 >>    PERCENT-SECONDS,    R  2  (5/10);
<<   104 >>    PERCENT-WRITE,      R  4  (5/10);
<<UN-REF>>     POTBLOCKMAX,        I  1  (5/10);
<<   106 >>    PRIMARYPATHNUM,     I  1  (5/10);
<<   107 >>    PROGRAM-NAME,       X 28  (5/10);
<<   108 >>    PROGRAM-VERSION,    X  8  (5/10);
<<   109 >>    PROG-MISCVAL1,      I  2  (5/10);
<<   110 >>    PROG-MISCVAL2,      I  2  (5/10);
<<   111 >>    PROG-MISCVAL3,      R  4  (5/10);
<<   112 >>    PROG-MISCVAL4,      R  4  (5/10);
<<   113 >>    PROG-MISCVAL5,      X 10  (5/10);
<<   114 >>    PROG-MISCVAL6,      X 10  (5/10);
<<   115 >>    PUTS,               I  2  (5/10);
<<   116 >>    RANDOMITY,          R  2  (5/10);
<<   117 >>    RECORDCODENUMBER,   I  1  (5/10);
<<   118 >>    RECOMMEND-TEXT,     X 78  (5/10);
<<   119 >>    SEARCHITEMTYPE,     X  2  (5/10);
<<   120 >>    SEARCHITEMLENGTH,   I  1  (5/10);
<<   121 >>    SEARCHITEMNAME,     X 16  (5/10);
<<UN-REF>>     SEARCHITEMNUMBER,   I  1  (5/10);
<<   123 >>    SET-SECTORSIZE,     I  2  (5/10);
<<   124 >>    SET-SECTORUSED,     I  2  (5/10);
<<   125 >>    SET-MISCVAL1,       I  2  (5/10);
<<   126 >>    SET-MISCVAL2,       I  2  (5/10);
<<UN-REF>>     SET-MISCVAL3,       I  2  (5/10);
<<   128 >>    SET-MISCVAL4,       R  4  (5/10);
<<   129 >>    SET-MISCVAL5,       R  4  (5/10);
<<UN-REF>>     SET-MISCVAL6,       R  4  (5/10);
<<   131 >>    SET-MISCVAL7,       X 10  (5/10);
<<   132 >>    SET-MISCVAL8,       X 10  (5/10);
<<UN-REF>>     SET-MISCVAL9,       X 10  (5/10);
<<   134 >>    SET-NAME,           X 16  (5/10);
<<   135 >>    SET-NUMBER,         K  1  (5/10);
```

```
<<  136 >>   SET-TYPE,           X  2  (5/10);
<<  137 >>   TIME-RUN,           I  1  (5/10);
<<UN-REF>>   TPI-ENABLED,        X  2  (5/10);
<<UN-REF>>   TPI-PRODUCT,        X 20  (5/10);
<<  140 >>   TRANSTIMEPUTS,      R  2  (5/10);
<<  141 >>   TRANSTIMEDELETES,   R  2  (5/10);
<<  142 >>   TRANSTIMEUPDATES,   R  2  (5/10);
<<  143 >>   UPDATES,            I  2  (5/10);
Sets:
                                          << Set No:  1 >>
Name:  CONTROL,Manual(5/10);
Entry: CONTROL-CODE(0),         <<   X  2 >>
       FORMATVERSION;           <<   X  8 >>
Capacity:          1 (1) ;
                                          << Set No:  2 >>
Name:  DATABASES,Automatic(5/10);
Entry: DATABASE-NAME(9);        <<   X 26 >>
Capacity:        301 (23) ;
                                          << Set No:  3 >>
Name:  DATASETS,Automatic(5/10);
Entry: SET-NAME(5);             <<   X 16 >>
Capacity:       1001 (47) ;
                                          << Set No:  4 >>
Name:  PROGRAMS,Automatic(5/10);
Entry: PROGRAM-NAME(1);         <<   X 28 >>
Capacity:        501 (51) ;
                                          << Set No:  5 >>
Name:  GLOBALDIAGS,Detail(5/10);
Entry: DATABASE-NAME(!DATABASES(DATE-RUN)),
                                <<   X 26 >>
       EXCLUSIVE,               <<   X  2 >>
       PROGRAM-VERSION,         <<   X  8 >>
       NUM-SETS,                <<   I  1 >>
       BLOCKMAX,                <<   I  1 >>
       NUM-ITEMS,               <<   I  1 >>
       MIN-BUFFSREQ,            <<   I  1 >>
       NUM-INDEXSETS,           <<   I  1 >>
       NUM-INDEXFILES,          <<   I  1 >>
       MIN-BUFFERS,             <<   I  1 >>
       NUM-DAYSHIST,            <<   R  2 >>
       NUM-CONCUSERS,           <<   I  2 >>
       NUM-WRITEACCESS,         <<   R  4 >>
       GLOBAL-MISCVAL1,         <<   I  2 >>
       GLOBAL-MISCVAL2,         <<   I  2 >>
       GLOBAL-MISCVAL3,         <<   I  2 >>
       GLOBAL-MISCVAL4,         <<   R  4 >>
       GLOBAL-MISCVAL5,         <<   R  4 >>
       GLOBAL-MISCVAL7,         <<   X 10 >>
       GLOBAL-MISCVAL8,         <<   X 10 >>
       DATE-RUN,                <<   X  8 >>
       TIME-RUN;                <<   I  1 >>
Capacity:        620 (31) ;
                                          << Set No:  6 >>
Name:  SETDIAGS,Detail(5/10);
Entry: DATABASE-NAME(!DATABASES(DATE-RUN)),
                                <<   X 26 >>
       EXCLUSIVE,               <<   X  2 >>
       NUM-ITEMS,               <<   I  1 >>
       NUM-PATHS,               <<   I  1 >>
       CAPACITY,                <<   I  2 >>
       NUM-ENTRIES,             <<   I  2 >>
       PERCENT-FULL,            <<   R  2 >>
       PERCENT-SECONDS,         <<   R  2 >>
       PERCENT-INEFF,           <<   R  2 >>
       RANDOMITY,               <<   R  2 >>
       ENTRY-LENGTH,            <<   I  1 >>
       BLOCKING-FACTOR,         <<   I  1 >>
       CLUSTERINGFACTOR,        <<   R  2 >>
       AVGE-SEC-CHAIN,          <<   R  4 >>
       LDEV-NUM,                <<   I  1 >>
       PERCENT-MESSY,           <<   R  2 >>
       PERCENT-DELETED,         <<   R  2 >>
       NUM-DELETED,             <<   I  2 >>
       NUM-SORTEDPATHS,         <<   I  1 >>
       PERCENT-MESSY2,          <<   R  2 >>
       PERCENT-MESSY3,          <<   R  2 >>
       SET-TYPE,                <<   X  2 >>
```

```
        SEARCHITEMNAME,              <<    X 16 >>
        SEARCHITEMTYPE,              <<    X  2 >>
        NUM-SECONDARIES,             <<    I  2 >>
        HIGHWATERMARK,               <<    I  2 >>
        MAX-FREESPACE,               <<    R  4 >>
        PERCENT-WRITE,               <<    R  4 >>
        NUM-CONTENTRIES,             <<    I  2 >>
        NUM-CONTBLOCKS,              <<    I  2 >>
        SEARCHITEMLENGTH,            <<    I  1 >>
        PRIMARYPATHNUM,              <<    I  1 >>
        SET-SECTORSIZE,              <<    I  2 >>
        SET-SECTORUSED,              <<    I  2 >>
        SET-MISCVAL1,                <<    I  2 >>
        SET-MISCVAL2,                <<    I  2 >>
        SET-MISCVAL4,                <<    R  4 >>
        SET-MISCVAL5,                <<    R  4 >>
        SET-MISCVAL7,                <<    X 10 >>
        SET-MISCVAL8,                <<    X 10 >>
        SET-NUMBER,                  <<    K  1 >>
        SET-NAME(DATASETS),          <<    X 16 >>
        DATE-RUN,                    <<    X  8 >>
        TIME-RUN;                    <<    I  1 >>
Capacity:      16000 (16) ;
                                              << Set No:   7 >>
Name:  PATHDIAGS,Detail(5/10);
Entry: DATABASE-NAME(DATABASES(DATE-RUN)),
                                     <<    X 26 >>
        MASTER-SETNAME(DATASETS),    <<    X 16 >>
        SEARCHITEMTYPE,              <<    X  2 >>
        SEARCHITEMLENGTH,            <<    I  1 >>
        AVERAGE-LENGTH,              <<    R  4 >>
        MAX-LENGTH,                  <<    I  2 >>
        MIN-LENGTH,                  <<    I  2 >>
        CHAIN-LENGTH,                <<    I  2 >>
        CHAIN-NUMBER,                <<    I  2 >>
        PATH-MISCVAL1,               <<    I  2 >>
        PATH-MISCVAL3,               <<    R  4 >>
        PATH-MISCVAL5,               <<    X 10 >>
        DETAIL-SETNAME(!DATASETS(PATH-NUMBER)),
                                     <<    X 16 >>
        PATH-NUMBER,                 <<    K  1 >>
        SEARCHITEMNAME,              <<    X 16 >>
        DATE-RUN,                    <<    X  8 >>
        TIME-RUN;                    <<    I  1 >>
Capacity:      12008 (19) ;
                                              << Set No:   8 >>
Name:  RECOMMENDS,Detail(5/10);
Entry: DATABASE-NAME(!DATABASES(SET-NUMBER)),
                                     <<    X 26 >>
        RECOMMEND-TEXT,              <<    X 78 >>
        RECORDCODENUMBER,            <<    I  1 >>
        DATE-RUN,                    <<    X  8 >>
        ACTIONED,                    <<    X  2 >>
        SET-NUMBER,                  <<    K  1 >>
        SET-NAME(DATASETS);          <<    X 16 >>
Capacity:       765 (17) ;
                                              << Set No:   9 >>
Name:  ANALYSEGLOBALS,Detail(5/10);
Entry: DATABASE-NAME(!DATABASES(DATE-RUN)),
                                     <<    X 26 >>
        PROGRAM-VERSION,             <<    X  8 >>
        NUM-DAYSHIST,                <<    R  2 >>
        MAX-LOGRECS,                 <<    I  2 >>
        MAX-PUTSPERDAY,              <<    R  2 >>
        MAX-DELETEPERDAY,            <<    R  2 >>
        MAX-UPDATEPERDAY,            <<    R  2 >>
        MAX-GETSPERDAY,              <<    R  2 >>
        NUM-OPENS,                   <<    I  2 >>
        NUM-CLOSES,                  <<    I  2 >>
        NUM-ABORTS,                  <<    I  2 >>
        NUM-ABNORMALENDS,            <<    I  2 >>
        PUTS,                        <<    I  2 >>
        DELETES,                     <<    I  2 >>
        UPDATES,                     <<    I  2 >>
        NUM-GETS,                    <<    I  2 >>
```

```
            NUM-DAYS,                     <<    R  2 >>
            MAX-CONUSERS,                 <<    I  2 >>
            GLOBAL-MISCVAL1,              <<    I  2 >>
            GLOBAL-MISCVAL2,              <<    I  2 >>
            GLOBAL-MISCVAL3,              <<    I  2 >>
            GLOBAL-MISCVAL4,              <<    R  4 >>
            GLOBAL-MISCVAL5,              <<    R  4 >>
            GLOBAL-MISCVAL7,              <<    X 10 >>
            GLOBAL-MISCVAL8,              <<    X 10 >>
            DATE-RUN,                     <<    X  8 >>
            TIME-RUN;                     <<    I  1 >>
Capacity:         630 (14) ;
                                                    << Set No:  10 >>
Name:  ANALYSESETS,Detail(5/10);
Entry: DATABASE-NAME(!DATABASES(DATE-RUN)),
                                  <<    X 26 >>
            CAPACITY,                     <<    I  2 >>
            NUM-ENTRIES,                  <<    I  2 >>
            SET-TYPE,                     <<    X  2 >>
            PUTS,                         <<    I  2 >>
            DELETES,                      <<    I  2 >>
            UPDATES,                      <<    I  2 >>
            NULL-UPDATES,                 <<    I  2 >>
            GETS,                         <<    I  2 >>
            TRANSTIMEPUTS,                <<    R  2 >>
            TRANSTIMEUPDATES,             <<    R  2 >>
            MAX-INCREASE,                 <<    I  2 >>
            IMPLIED-UPDATES,              <<    I  2 >>
            NUM-DAYS,                     <<    R  2 >>
            SET-MISCVAL1,                 <<    I  2 >>
            SET-MISCVAL2,                 <<    I  2 >>
            SET-MISCVAL4,                 <<    R  4 >>
            SET-MISCVAL5,                 <<    R  4 >>
            SET-MISCVAL7,                 <<    X 10 >>
            SET-MISCVAL8,                 <<    X 10 >>
            SET-NUMBER,                   <<    K  1 >>
            SET-NAME(DATASETS),           <<    X 16 >>
            DATE-RUN,                     <<    X  8 >>
            TIME-RUN;                     <<    I  1 >>
Capacity:       15008 (14) ;
                                                    << Set No:  11 >>
Name:  ANALYSEPROGS,Detail(5/10);
Entry: DATABASE-NAME(!DATABASES(DATE-RUN)),
                                  <<    X 26 >>
            PUTS,                         <<    I  2 >>
            DELETES,                      <<    I  2 >>
            UPDATES,                      <<    I  2 >>
            NULL-UPDATES,                 <<    I  2 >>
            GETS,                         <<    I  2 >>
            TRANSTIMEPUTS,                <<    R  2 >>
            TRANSTIMEDELETES,             <<    R  2 >>
            TRANSTIMEUPDATES,             <<    R  2 >>
            IMPLIED-UPDATES,              <<    I  2 >>
            NUM-DAYS,                     <<    R  2 >>
            PROG-MISCVAL1,                <<    I  2 >>
            PROG-MISCVAL2,                <<    I  2 >>
            PROG-MISCVAL3,                <<    R  4 >>
            PROG-MISCVAL4,                <<    R  4 >>
            PROG-MISCVAL5,                <<    X 10 >>
            PROG-MISCVAL6,                <<    X 10 >>
            PROGRAM-NAME(PROGRAMS),       <<    X 28 >>
            DATE-RUN,                     <<    X  8 >>
            TIME-RUN;                     <<    I  1 >>
Capacity:        6006 (14) ;
                                                    << Set No:  12 >>
Name:  ANALYSEHOURS,Detail(5/10);
Entry: DATABASE-NAME(!DATABASES(DATE-RUN)),
                                  <<    X 26 >>
            NUM-DAYS,                     <<    R  2 >>
            OPENS,                        << 24 I  2 >>
            CLOSES,                       << 24 I  2 >>
            ABORTS,                       << 24 I  2 >>
            ABNORMAL-ENDS,                << 24 I  2 >>
            NUM-PUTS,                     << 24 I  2 >>
            NUM-DELETES,                  << 24 I  2 >>
            NUM-UPDATES,                  << 24 I  2 >>
            HOUR-MISCVAL1,                << 24 I  2 >>
```

```
        HOUR-MISCVAL3,              << 24 R  4 >>
        DATE-RUN,                  <<    X  8 >>
        TIME-RUN;                  <<    I  1 >>
Capacity:        620 (4) ;

                                        << Set No:  13 >>
Name:  ACTIONS,Detail(5/10);
Entry: DATABASE-NAME(!DATABASES(DATE-RUN)),
                                   <<    X 26 >>
        PROGRAM-NAME,              <<    X 28 >>
        OLDVALUE,                  <<    X 10 >>
        NEWVALUE,                  <<    X 10 >>
        SET-NAME,                  <<    X 16 >>
        SET-NUMBER,                <<    K  1 >>
        DATE-RUN;                  <<    X  8 >>
Capacity:        260 (26) ;
END.
```

NB. The initial capacities are set as above, however the user may modify these using FLEXIBASE at any time. However, please note that if a dataset fills up during a Diagnose or Analyze run the remainder of the run will continue without storing any information in FXSTAT.PUB.HQ. An error message will be displayed on FXLIST if this condition is encountered, and a capacity change executed. The user should regularly check on the capacities of these sets to ensure that there is sufficient room for the next few runs of Diagnose and Analyze. In particular the exact capacities required for the PATHDIAGS and SETDIAGS sets, for example, will depend upon the mix of databases the user runs Diagnose and Analyze on, and will vary from user to user.

Most of the above data items refer to the Diagnose and Analyze statistics, the definitions of which can be found elsewhere in this manual. Some items (specifically those which are unreferenced or those with "MISCVAL" in their names) are present for future expansion and are not currently used. DATERUN is in the format "YYYYMMDD" (e.g. 19940430) and corresponds to the date at the time Diagnose was begun on this database. TIMERUN is in the form of an integer number "HHMM" in 24 hour format, thus a value of "130" represents 1:30am and a value of "2358" represents 23:58 or 11:58pm.

The RECOMMENDS set contains the set of recommendations last issued for this database, and are cleared out at the start of the next run. RECORDCODENUMBER is an internal reference number for the recommendation. ACTIONED is a field that is not used at present.

The ANALYSEHOURS set contains data items whose 24 sub-items represent the 24 hours of the day, sub-item 1 relating to midnight-1am, 2 representing 1am-2pm etc.

This database will be created as part of the installation procedure of releases E.04.01 or above. If the database FXSTAT.PUB.HQ already exists it will not be overwritten in subsequent release.

SECTION 24 ℂ FXUTIL CONFIGURATION UTILITY

Introduction

The FXUTIL Configuration utility allows you:

to modify the performance and formatting parameters for the the ANALYZE and DIAGNOSE functions

to create utility command files in IMAGESQL or ISQL formats, which can be used in modifying the configuration of IMAGE/SQL ATTACHed databases.

To enable flexibility in terms of output and parameters from the ANALYZE and DIAGNOSE modules, FXUTIL maintains a file in PUB.HQ containing the following parameters for the ANALYZE module:
- Output device
- Output page width
- Output page length
- Date format
- Blank entry suppression
- Lotus output option
- Number of days between statistics updates

and the following parameters for the DIAGNOSE module:
- Max Master Percent Full
- Max Detail Percent Full
- Max Secondary Percentage
- Max Inefficient Percentage
- Target Master Percent Full
- Target Detail Percent Full
- Target Secondary Percentage
- Target Inefficient Percentage
- Lotus output option
- Max Percentage Messy
- Number of days between statistics updates
- Number of days allowed for capacity increase recommendations

To run this utility one must have read and write access to the PUB.HQ group. A file containing default parameter values is provided on your installation tape.

To run this program select option 20 from the main FLEXIBASE menu (MPE Commands) and enter "FXUTIL." This will run the program FXUTIL.PUB.HQ and open the file FXANCFG.PUB.HQ. The utility contains options for configuring the ANALYZE and DIAGNOSE module parameters, and for generating IMAGE/SQL configuration command files.

The following is the FXUTIL menu:

```
<><><>  FXUTIL  <><><>       Version  E.05.01   (c) Sector 7 Software 1989-1996
                                sold under license by PROACTIVE SYSTEMS worldwide
                                             (THU, JAN 11, 1996,  1:21 AM)

1 CONFIGURE   output parameters for ANALYZE module
2 CONFIGURE   run-time parameters for DIAGNOSE Module
3 MAKE        ASCII file from Re-Attachment QR File
4 MAKE        QR and QL Re-attachment Files for a database
5 MAKE        QL Re-attachment File for a single set

CHOOSE (1-5)
```

ANALYZE Parameters

Select Option one from the menu.

The current parameters for output from ANALYZE are now displayed, one at a time. The user is then asked to enter the new value. Entering return leaves the current value as it was. Any other value is validated and accepted if it is within the relevant constraints.

```
Current Output Device configured = LASER
Enter required Output Device >>
```

Enter a valid device name, including $STDLIST, $NULL etc., to which output from ANALYZE will be directed. If the device name is DISC then the user will be prompted for a disc file name at run time. This device may be overridden by a file equation; however if DISC is redirected to LP the program will still ask for a disc file name. To avoid this, re-run FXUTIL and modify the contents of the file rather than use a file equation. The default for this option is device LP.

```
Current Page Width configured = 178
Enter required Page Width >>
```

Enter a number between 1 and 256, specifying the maximum number of characters per line for output from ANALYZE. This value may not be overridden by a file equation. The default is 132.

```
Current Page Length configured = 66
Enter required Page Length >>
```

Enter a number between 1 and 99 to specify the number of lines per page for output from ANALYZE. The default is 66.

```
Current Date Format configured = 0
Enter required Date Format (0-2) >>
```

Enter the required date format for all input and output of dates in ANALYZE. Enter a number between 0 and 2 to select one of the following formats:

0	dd/mm/yy	for example, 31/12/89.
1	mm/dd/yy	for example, 12/31/89.
2	ddmmmyy	for example, 31DEC89.

The default for this parameter is 2.

```
Current configured Blanks option = YES
Enter required Blanks option (YES/NO) >>
```

Enter YES if blank entries in the log file (for example, DBUPDATE entries which have not changed any data item values at all C NullUpdates) are to be ignored in the Audit Trail and Enquire options of ANALYZE. The logging process logs DB calls even if no actual data has changed or been added, for example DBUPDATE calls which actually change no data, or DBBEGIN calls with no text. These entries

will not be displayed in the Audit trail and Enquiry options if YES is selected in this option. Otherwise these calls will appear, but will use up paper. The default for this option is YES.

```
Current Lotus Output Option configured =

Enter Lotus Output Option (YES/NO)      >>
```

The output of the ANALYZE and DIAGNOSE modules can also be stored in a Lotus-123 compatible "Comma Separated Variable" file. This file can then be imported into any Lotus-123 compatible spreadsheet for further processing and for historical comparisons. Enter YES to activate this option, or NO (default) to disable it. This file will be created in addition to the normal ANALYZE and DIAGNOSE output, and you will be asked for the file's name at run time.

```
Enter required Num days between Updates  >>
```

Enter the minimum number of days to allow between updating the FXSTAT.PUB.HQ statistics data base. This database will store data for each set and path for each complete run of Analyze. A small value will result in this database becoming full quickly, large values will reduce the effectiveness of the trends available from the data. The default value is 30.

◇◇◇◇◇ End of Configuring Parameters ◇◇◇◇◇

DIAGNOSE Parameters

All users do not have the same database capacity and performance requirements. Where disc space is tight a user may try and keep datasets' capacities such that the datasets are at 90% full or more. Because of these differing requirements DIAGNOSE allows you to configure its decision-making parameters.

DIAGNOSE uses the following default parameters in its recommendations and diagnoses:

Max Master Percent Full	(default value = 80)
Target Master Percent Full	(default value = 70)
Max Detail Percent Full	(default value = 85)
Target Detail Percent Full	(default value = 70)
Max Secondary Percentage	(default value = 30)
Target Secondary Percentage	(default value = 26)
Max Inefficient Percentage	(default value = 24)
Target Inefficient Percentage	(default value = 20)
Max Percent Messy	(default value = 15)
Current Lotus Output Option	(default value = NO)

When first installed the value of these parameters are set to their default values. However, the user may configure any of the above values using the FXUTIL utility (within certain common sense restrictions, for example the Target Master Set Percent Full cannot be higher than the maximum).

Unless there are good local reasons to change these values we recommend they be left at their defaults. They have been chosen to allow early detection of critical conditions and to recommend sufficient increases to minimize future repetitions of those conditions. However, because there is some built-in latitude in these defaults it may be more appropriate to establish some local parameters.

```
Enter Num Days between Stat DB Updates  >>
```

Enter the minimum number of days to allow between updating the FXSTAT.PUB.HQ statistics data base. This database will store data for each set and path for each complete run of Diagnose. A small value will result in this database becoming full quickly, large values will reduce the effectiveness of the trends available from the data. The default value is 30.

```
Enter Num Days for Capacity Increases   >>
```

Enter the number of days to allow for when calculating new Capacities (if needed). Diagnose will calculate the appropriate Capacity based upon the normal parameters, but will also ensure that this allows for the average number of entries that will be added in this number of days. The average number of entries is calculated using data from the FXSTAT.PUB.HQ database. The default value is 60.

QR AND QL Files

See Section 18.2 on support for IMAGE/SQL for an explanation of the uses for QR and QL files. By using FXUTIL, you can generate these files for your own use.

Option 3 on the FXUTIL menu will copy the QR file as a non-privileged text file. The file name is FXATCCFG. Note that this is a temporary file, and will be lost when your session ends. This is designed to minimize the risk of files containing IMAGE passwords being left on the system.

Option 4 on the FXUTIL menu will generate the pair of files at any time for a database without using TRANSMUTE.

The DBA thus has the ability to perform a full automatic re-attach in the event of an accidental DETACH. The QR file would be used as input to IMAGESQL and the QL file would be used as input to ISQL.

Option 5 on the FXUTIL menu will generate an SQL file relating to a single set from a database. If this is done before the SPLIT or UPDATE is performed, then this file can be used to recreate the dropped SQL objects after the SPLIT or UPDATE, by using the file as input to ISQL.

APPENDIX A Ⅽ INSTALLATION & DEMONSTRATION

INSTALLATION OF HP3000 SOFTWARE

To install the FLEXIBASE software on your system follow the instructions below.
- Log on as MANAGER.SYS and restore the file JFLEXINS from the tape supplied.
- Use EDITOR to insert the MANAGER.SYS password into line 1 of that file.
- Keep the file JFLEXINS, exit from EDITOR and introduce it as a job using the command :STREAM JFLEXINS.

NOTE: If you already have FLEXIBASE installed on your system, you may need to modify the second !JOB command to include passwords for DPCONT.HQ.

The installation is carried out by two jobs:
- the first (called JFLEXINS) will build a suitable account structure (in account HQ) and restore the product files (a second tape mount will be necessary), and then stream a second job
- the second job (also called JFLEXINS) will run the FXUPGRAD program to install or upgrade the statistics database FXSTAT and the FLEXIBASE configuration file, and purge any files no longer required by new releases of the product.

When the second job has finished, the FLEXIBASE system will be installed and ready for use (unless a failure has occurred Ⅽ check the STDLIST if necessary). The following files will be restored.

FLEXIBAS: The FLEXIBASE menu program.

TRANSMUT, TRANSMUX, FXDMAPPP, FXDMAPPX: The TRANSMUTE database transformation programs.

FXTPOSE, FXTPOSEX: The TRANSPOSE programs for changing key item values.

FLEXISET, FLEXISEX: The programs containing routines for dealing with sets.

FXDECO, FXDECOX: The programs for decompiling root files into schemas.

FLEXICAP, FLEXICAX: The capacity change programs.

DIAGNOSE, DIAGNOSX: The diagnosis programs.

COPYBASE, COPYBASX: The database copy programs.

FXCREATE, FXCREATX: The database creation programs.

FXTSFER, FXTSFERX: The TRANSFER programs.

FXSORTA, FXSORTB, FXSORTAX, FXSORTBX: Custom sort programs used in REPACKing.

FLEXIMAP, FLEXIMAX; FLEXMAP2, FLEXMAPX: The dataset repack programs.

FXINTCHK, FXINTCHX: The data integrity checking programs.

PFXHOSTI: Server software to work in conjunction with the PC client.

TMDEMO1: Database schema file for the database FIRST, used in the demonstration runs.

TMDEMO2: Database schema file for the database SECOND, used in the demonstration runs.

FXANALYS, FXANALYX: The log file analysis programs.

FXANCFG: The configuration file for ANALYZE and DIAGNOSE.

FXUTIL: Configurator program for ANALYZE and DIAGNOSE.

FXUPGRAD: Upgrade program for statistics database (only used during installation).

FXRENAME, FXRENAMX: The database rename programs.

FXARCHIV, FXARCHIX: The ARCHIVE function programs.

FL1, FL2: Data files that are used to build a test database named FIRST.

FINIT: A program used to load the test database.

TMXEQ1: QUERY XEQ file for database FIRST.

TMXEQ2: QUERY XEQ file for database SECOND.

TMMESSD@: The FLEXIBASE message files. File TMMESSDA is the English version with other suffixes for other languages.

FXSTATSC: Schema file for SXSTAT statistics database.

CFLEX etc: The FLEXIBASE manual in FANTASIA format.

DBCHECK: Unsupported HP utility to check the internal structure of an IMAGE database.

TDBCHECK: TurboIMAGE version of DBCHECK.

HQFX: A demonstration database for the ANALYZE module.

HQFXLOG: A demonstration logfile for the ANALYZE module.

HQFXFX: A Database History File for collecting statistics.

FXUPGRAD: program to create and upgrade the FXSTAT database.

SQLMAIN, SQLRTNXL: XL and program to support IMAGESQL log file generation

S7CAT000, PFXIDEP: program to support SQL object generation

Note that on MPE/V, FLEXIBASE requires that your operating system is configured with a Maximum Code Segment Size of 16384. If you currently have a lesser figure configured then certain FLEXIBASE functions may fail with a "Hard Load Error" message. It is suggested that you reconfigure your operating system using a SYSDUMP/UPDATE dialog to overcome this problem C for most HP3000 installations there is little point in having a lesser figure configured.

FLEXIBASE is supplied in both MPE/V and MPE/iX modes. To save disc space, you may want to delete unnecessary programs from your system. If you are running on MPE/V, then you should delete all files of type NMPRG, NMXL, and MGCAT. If you are running under MPE/iX, then delete all files of type PROG, with the exception of the menu program, FLEXIBAS.PUB.HQ, and the server program PFXHOSTI.

To use the French, German or Spanish language versions of the software, refer to Appendix K.

INSTALLATION OF PC SOFTWARE

The PC Software is supplied on a diskette. The procedure is the same whether this is a first time installation of FLEXIBASE/IMAGE, or an update to a previous version.

This software is written in Visual Basic 3.0, and uses the standard Microsoft installation procedure for Visual Basic programs. You should be aware of the following implications:

- The Microsoft setup program will need to install the Visual Basic run-time procedure library in \WINDOWS\SYSTEM. It is therefore important that this directory should be available for <u>write</u> access when you are doing the installation (this may not ordinarily be the case if you are running Windows from a server, or Windows for Workgroups).
- FLEXIBASE uses a number of Visual Basic common modules (.VBX and .DLL files) as well as the Graphics Server GSW.EXE. Following Microsoft's recomendation, these will be installed in \WINDOWS\SYSTEM. If earlier versions of these files are already installed, then they will be overwritten with the latest versions from the installation diskette. If later versions are already installed, then they will not be overwritten. In this way, at the end of the installation process, the latest versions of the Visual Basic common modules will be installed on your system. Since Visual Basic 3.0 is upwards compatible, this installation process should not cause any problems for existing software.

The following instructions assume that you insert the Setup diskette into drive A, but if you wish to use another drive simply substitute the letter of that drive for "**A:**".

- If Windows is not already running, type **win** at the DOS prompt.
- **Insert** the FLEXIBASE/IMAGE Setup disk in drive A, and close the drive door.
- In the **Program Manager** window, Click on **File** in the menu-bar, then select **Run.**
- Type **A:SETUP** and then click on the **OK** button.
- To accept the path that Setup proposes, click on the **Continue** button - or -
- To choose your own directory, type a **new path** in the "Install" field, and then click on the **Continue** button. (You are recommended to accept the default path.)
- A progress histogram will inform the user of percentage completion of the setup process.
- A dialog box will inform you that installation is complete. Click on the **OK** button.
- Setup will ask if you want to view the README.WRI file, which contains details about this release. Click on the **Yes** or **No** button.
- A FLEXIBASE group window is now visible within program manager, containing the following icons:-
    - FLEXIBASE/IMAGE Trends: the software icon
    - Help on FLEXIBASE: this provides a direct access to the FLEXIBASE on-line help.
    - The Readme file accompanying this release of the software
    - Three MS-Write icons which provide access to detailed information on configuring connections between the PC and the HP3000.

    Any of these icons can be moved to other programs groups.

- Click on the FLEXIBASE/IMAGE icon when you require to run the software.

When you run the FLEXIBASE software and use <u>C</u>onnections, <u>O</u>pen in the menu, you will see four connections already configured. These connections are provided as templates, to show how to set up connections in FLEXIBASE. Note that you should configure your own connections <u>under different names</u>, otherwise they will be overwritten when you install future versions of the software (note that the LAN connection will only be available when Telnet is directly supported by MPE/iX).

DEMONSTRATION

To do a simple capacity change using FLEXIBASE, proceed as follows:

Log on as DPCONT.HQ.

Type "RUN FLEXIBAS.PUB.HQ" to run the FLEXIBASE software.

Select option 5 from the menu. This runs the CAPACITY function which will prompt as follows:

```
Do you want to run Interactively now or create a
streamed Job file.
     Enter I or J >>
```

Reply I for interactive.

```
Database name >>
```

Reply FIRST

A slight pause occurs at this point while FLEXIBASE checks the database specified.

```
Set name >>
```

Reply CUSTOMER

Note: if you didn't know the set name the entry of a question mark (?) would have given us a list of sets in the database. A question mark can be entered in response to any prompt to invoke the on-line, context sensitive, help facility.

FLEXIBASE will now display the current number of entries (13) and the existing capacity (19).

```
Enter new capacity (or change %) >>
```

Enter +20%

The program will display the proposed new capacity.

```
Do you want to proceed Ȼ Enter Y or N >>
```

Enter Y

The progress of the capacity change (as a percentage complete) is displayed followed by the message "CAPACITY CHANGE COMPLETE". Another set name and database name will then be prompted for. Press return in each case to exit from the function.

DEMONSTRATION OF DIAGNOSE

As a demonstration of the analytic capabilities of FLEXIBASE we will execute the DIAGNOSE function. In this example we will run in job mode to show how easy it is to create standard functions that can be saved for repeated running.

Select option 2 from the menu after logging on as DPCONT.HQ and entering ":RUN FLEXIBAS.PUB.HQ". This runs the DIAGNOSE function which prompts as follows:

```
Do you want to run Interactively now or create a
streamed Job file.
     Enter I or J >>
```

Reply J for job mode.

The program will then prompt for a User Password, Group Password and Account Password ℂ press Return in each case unless you have placed any passwords on the HQ account.

```
Running FLEXIBASE in this mode will create an MPE job stream file.
The default filename is FLEXIJOB, you may select another name.
Enter name of job file [return=FLEXIJOB] >>
```

You may select a specific job file name , but for this demo simply pressing Return will use the default file FLEXIJOB. If this file exists you will be asked if the file should be replaced, appended to or the name re-entered. For this demo select R for replace.

```
FLEXIJOB.PUB.HQ jobfile will be newly created

Database name >>
```

Reply FIRST

```
Enter Password >>
```

Reply ";"

```
Enter Set Name (or @ for all sets) >>
```

Enter @

```
Do you want the summary or detail report on $STDLIST >>
```

Reply S.

The program will then prompt for another database ℂ press Return once to exit. The program will then ask when the job file should be streamed. You have the option of streaming the job now, at a future date/time or saving the file. Select S to stream now to initiate the diagnosis run. Press Return to go back to the main menu again. After the job has completed, a full analysis of the test database will be printed on your system line printer. Depending on which version of MPE you are running, you should see a recommendation either to REPACK the TRANSACTIONS set and change its blocking factor, or to change the blocking factor and increase the BLOCKMAX for the database as a whole. A recommendation to change its capacity should also be present.

DEMONSTRATION OF ANALYZE

The installation tape also contains a demonstration database and logfile for use with the ANALYZE module. The database is called HQFX.PUB.HQ; the logfile is HQFXLOG.PUB.HQ. The logging process used to create this logfile was called FXLOG and had a password of TEST. To run the demonstration follow the example below, entering the values indicated, after logging on as DPCONT.HQ.

Select option 4 from the FLEXIBASE menu. Reply I for interactive mode and then proceed as shown by the dialog below. Output will appear on the system line printer

```
<><> ANALYZE/3000 <><>    Version D.00.01      (c) Sector 7 Software 1989
                     sold under license by PROACTIVE SYSTEMS worldwide
                                        (SUN, OCT  1, 1989,  2:50 PM)
Enter Database name >> HQFX.PUB.HQ
Enter Password >> ;


Database HQFX.PUB.HQ has Logid of FXLOG
Enter Logid Password >> TEST

Enter Logfile Name >> HQFXLOG.PUB.HQ
Logfile HQFXLOG.PUB.HQ contains        48 log records
Logfile starts at 21:41 27/09/89
and runs until    21:25 28/09/89

You can analyze the above Database and Logfile in three different ways:
---------------------------------------------------------------------
 A - Audit Trail Option    : A sorted trail of selected log entries
 E - Logfile Enquiry Option : Full details of selected log entries
 S - Statistics Option     : Logfile and historical statistics
---------------------------------------------------------------------
Enter one of the above letters to choose an option,  or press RETURN to
end the analysis of this Database/Logfile combination.

Select Option (A,E or S) >> S

OK to Update Database History File (Y/N) >> Y

<><><><><><><><><>  ANALYZE/3000 - Log Statistics Option <><><><><><><><>

Reading Log File...................
 21:41 27/09/89 <><><>   Logid FXLOG    Started    <><>
 00:10 28/09/89 <><><>   Logid FXLOG    Restarted  <><>
      36 entries selected

Producing User & Set Statistics.....
Reading database history file.......
Logfile has already been added to history file.

<><><><><><><><><><>   End of Log Statistics Option  <><><><><><><><><>
```

```
Select Option (A,E or S) >> A

<><><><><><><><><>   ANALYZE/3000 - Audit Trail Option   <><><><><><><><>

Sort by Users, Ldev Nos or Processes   (U,L or P) >> P
Print only DBBEGINs, Ends & Memos with Text (Y/N) >> N

Selection Criteria:
Enter Dataset Name  >> (enter return)
Enter User Name     >> (enter return)
Enter Program Name  >> (enter return)
Enter Ldev Number   >> (enter return)
Enter start time/date >> (enter return)
Enter end   time/date >> (enter return)

Proceed with this selection (Y/N) >> Y

Reading Log File...................
 21:41 27/09/89 <><><>   Logid FXLOG    Started    <><>
 00:10 28/09/89 <><><>   Logid FXLOG    Restarted  <><>
      30 entries selected
Sorting Output.....................
Printing Audit Trail...............
<><><><><><><><><>      End of Audit Trail Option     <><><><><><><><><>
```

```
Select Option (A,E or S) >> E

<><><><><><><><>   ANALYZE/3000 - Log Enquiry Option   <><><><><><><><>
Selection Criteria:
Enter Dataset Name  >> CUSTOMER
Enter Program Name  >> TRANSPOS.PUB.HQ
Enter User Name     >> (enter return)
Enter Ldev Number   >> (enter return)
Data Item Condition >> (enter return)
Enter Start Time/Date >> (enter return)
Enter End   Time/Date >> (enter return)

Proceed with this selection (Y/N) >> Y

Reading Log File...................
 21:41 27/09/89 <><><>   Logid FXLOG    Started    <><>
 00:10 28/09/89 <><><>   Logid FXLOG    Restarted  <><>
        2 entries selected

<><><><><><><><><>   End of Log Enquiry Option   <><><><><><><><><><>
```

```
Select Option (A,E or S) >> E

<><><><><><><><>   ANALYZE/3000 - Log Enquiry Option   <><><><><><><><>

Selection Criteria:
Enter Dataset Name  >> CUSTOMER
Enter Program Name  >> (enter return)
Enter User Name     >> (enter return)
Enter Ldev Number   >> (enter return)
Data Item Condition >> CREDIT-LIMIT<>0
Enter Start Time/Date >> (enter return)
Enter End   Time/Date >> (enter return)

Proceed with this selection (Y/N) >> Y

Reading Log File...................
 21:41 27/09/89 <><><>   Logid FXLOG    Started    <><>
 00:10 28/09/89 <><><>   Logid FXLOG    Restarted  <><>
        1 entries selected

<><><><><><><><><>   End of Log Enquiry Option   <><><><><><><><><><>

Select Option (A,E or S) >> (enter return)

Enter Database name >> (enter return)

END OF PROGRAM
:
```

DEMONSTRATION OF TRANSMUTE IN SCHEMA MODE

To demonstrate TRANSMUTE we have provided you with a database plus two schemas (the "original" form which matches the database named "FIRST," and a modified form). These schemas differ in several respects and the TRANSMUTE program will process the records in the database and convert their format to match the new schema. The contents of the two schema files are shown below:

```
DATABASE FIRST (SCHEMA FILE TMDEMO1)

Begin Database FIRST;
Passwords:
     5 READER;
    10 WRITER;

Items:
    ACCOUNT-NO,              J (5/10);
    NAME,                    X40(5/10);
    ADDRESS,                 X30(5/10);
    CREDIT-LIMIT,            J (5/10);
    DATE,                    X6(5/10);
    AMOUNT,                  J (5/10);
    TYPE,                    U4(5/10);

Sets:
Name:  CUSTOMER,Manual(5/10);
Entry: ACCOUNT-NO(1),
       NAME,
       ADDRESS,
       CREDIT-LIMIT;
Capacity: 19;
Name:  TRANSACTIONS,Detail(5/10);
Entry: ACCOUNT-NO(CUSTOMER),
       DATE,
       AMOUNT,
       TYPE;
Capacity: 250(13);
End.
```

```
MODIFIED DATABASE (SCHEMA FILE TMDEMO2)

Begin Database SECOND;      <<Changed from FIRST>>
Passwords:
     5 READER;
    10 WRITER;

Items:
    ACCOUNT-NO,              X4(5/10); <<Changed from J>>
    NAME,                    X40(5/10);
    ADDRESS,                 3X30(5/10); <<Changed from X30>>
    CREDIT-LIMIT,            J (5/10);
    DATE,                    X6(5/10);
    AMOUNT,                  R4(5/10); <<Changed from J>>
    TYPE,                    U4(5/10);

Sets:
Name:  CUSTOMER,Manual(5/10);
Entry: ACCOUNT-NO(1),
       NAME,
       ADDRESS,             <<2 Sub-items Added>>
       CREDIT-LIMIT;
Capacity: 29;          <<Changed from 19>>
Name:  TRANSACTIONS,Detail(5/10);
Entry: ACCOUNT-NO(CUSTOMER),
       DATE,
       AMOUNT,
       TYPE;
Capacity: 250(13);
End.
```

As you can see from examining the above, the following changes have been made to the second schema:

The ACCOUNT-NO item has been changed from a numeric J1 item to a character X4 item.

The ADDRESS item has been expanded from a single item to a multiple occurrence item.

The AMOUNT item has been changed from a numeric J1 item to an R4 item.

The capacity of the CUSTOMER set has been increased to 29.

The database is also renamed.

Execute the following instructions to run the demonstration:

Log on as DPCONT.HQ (the creator of the database).
Type RUN FLEXIBAS.PUB.HQ and select option 17 from the menu.
This runs the TRANSMUTE function which will ask a few questions.
Answer as follows:

```
Do you want to run Interactively now or create a
streamed Job file.
     Enter I or J >>
```

Reply I for interactive.

```
Old Database name >>
```

Reply FIRST

A slight pause occurs at this point while FLEXIBASE checks the database specified.

```
Enter New Schema filename [or return = MENU for changes] >>
```

Reply TMDEMO2

```
Capacities to be determined by Schema or left unchanged? (S/L) >>
```

Reply S

```
Blocking-factors to be determined by Schema or left unchanged? (S/L) >>
```

Reply S

A pause occurs at this point while FLEXIBASE checks the specified schema and builds various work files.

A prompt then appears which asks whether leading zeros on the ACCOUNT-NO item should be output as zeros or spaces in the new version of the item.

```
Enter Z or S >>
```

Reply S for Spaces

A warning message is displayed about uniqueness of keys which can be ignored. A message then appears asking whether the new sub-item should come Before or After existing sub-items.

```
Enter B or A >>
```

Reply A

A prompt requesting the number of decimal places for the AMOUNT item (which has changed in type) is now displayed.

```
Enter number >>
```

Reply 0 for the number of decimal places.

A definition of all the changes that have been identified is now displayed and the program requests confirmation to proceed. Note that because the CUSTOMER set has changed in content (and hence entry

length), reblocking of that set will take place and will be indicated in the list of changes. Although the TRANSACTION set has also changed in content, its blocking factor is dictated in the schemas.

```
Do you wish to proceed with the physical changes as above?
     Enter Y or N >>
```

Reply Y

The program will now transform the database FIRST with respect to the modified schema file TMDEMO2 into a database called SECOND. A listing of all the prompts, user responses and changes made will appear on the system line printer.

If you wish to see the contents of the new database then enter function 20 (MPE COMMANDS) and run the QUERY program (simply type QUERY in response to the prompt) and enter the following:

>XEQ TMXEQ2

This will report all the entries that now exist in the database SECOND to the line printer. Press return afterwards to go back to the main menu.

DEMONSTRATION OF TRANSMUTE IN INTERACTIVE MODE

This demonstration is a continuation of the previous one, and therefore assumes that the modifications contained in the Schema Mode demonstration of TRANSMUTE have already been applied. It will guide you through the steps to be taken to create a new item in the SECOND database, called PREV-LIMIT, and add it to the CUSTOMER dataset.

Begin by choosing the TRANSMUTE function as before, but this time give SECOND as the database name, and reply with return to the request for a schema file name. You will then be shown the Interactive TRANSMUTE menu. The different options on the menu are chosen by entering the number corresponding to the option that you want to execute. When you have finished making modifications, enter P to process changes: you can abandon your changes at any time by entering Q at the main menu prompt. The only exceptions to this rule are the rename options, which are incompatible with any others, and are executed immediately. In the following dialog, your replies are in bold characters.

First enter 21 to create a new item in the database:

```
CREATE NEW ITEM in database:

Enter NAME for new item [return=no more] >> PREV-LIMIT

Enter number of SUB-ITEMS if compound [return=simple item] >>
Enter Item DATA-TYPE >> J
Enter item LENGTH >> 1

ITEMS in database are:
1 ACCOUNT-NO   2 NAME     3 ADDRESS    4 CREDIT-LIMIT
5 DATE     .... etc

Insert BEFORE which item [or return=END of itemlist] >> 5

Security READ CLASSES:
Enter User-Class-No (0-63) [Return if no more] >> 5
Enter User-Class-No (0-63) [Return if no more] >>

Security WRITE CLASSES:
Enter User-Class-No (0-63) [Return if no more] >> 10
Enter User-Class-No (0-63) [Return if no more] >>

CREATE NEW ITEM in database:

Enter NAME for new item [return=no more]
```

Press return. You will then be returned to the options menu. Now select option 1 to add the new item to a set. The dialog continues as follows:

```
 ADD ITEM into Set
Available SETS in database are:
1 CUSTOMER     2 TRANSACTIONS

Select the Set >> 1
                 CUSTOMER

ITEMS in database are:
1 ACCOUNT-NO   2 NAME     3 ADDRESS 4 CREDIT-LIMIT
5 PREV-LIMIT   6 DATE     7 AMOUNT  8 TYPE

Select the item to be added into the set [return=no more] >> 5

ITEMS in this set are:
1 ACCOUNT-NO   2 NAME     3 ADDRESS 4 CREDIT-LIMIT

Insert BEFORE which item [or return=END of set] >>
ITEMS in database are:
1 ACCOUNT-NO   2 NAME     3 ADDRESS 4 CREDIT-LIMIT
5 PREV-LIMIT   6 DATE     7 AMOUNT  8 TYPE

Select the item to be added into the set [return=no more] >>
```

Once you have returned to the main option menu, you should select P to process the changes you have made. TRANSMUTE will then identify the modifications that will be applied to the existing database, and ask if you wish to proceed with them.

Note that you would need to re-install the FLEXIBASE software if you want to run through the demonstration again, since changes have been made to the data during the demonstration. After running the above tests, you could then try the software on one of your own databases. You may find it worthwhile to read more of this manual before attempting to run the software on your own files Ȼ however FLEXIBASE will generally prompt you for the necessary information.

One useful test of the capability of FLEXIBASE is to run the CHECK function on your operational databases to try and pick up logical errors in them that you may not be aware of.

We recommend that you use STORE or DBSTORE to back up your databases before using any of the database updating functions of the FLEXIBASE software on them (if you make a mistake in any new

schema file you supply or the responses you make to the FLEXIBASE program then you may need to restore the old versions of the databases).

Note that the FANTASIA REPORTER software is not included on FLEXIBASE demonstration installation tapes. Contact Open Seas or your local distributor for further information or a demonstration version of this optional module if you require it.

APPENDIX B C MANUAL PRINTING

The FLEXIBASE Technical Reference Manual is provided in both printed form and computer readable form. You can print more copies of the manual if you have a copy of the FANTASIA software and a LaserJet printer. FANTASIA is a typesetting, graphics and forms software package for HP 3000 users marketed by Open Seas. The manual file is CFLEX.PUB.HQ.

Also provided with FLEXIBASE is a Guide for the New User, which can be printed in the same way as the Technical Reference Manual. The manual file is CFLEXGUD.PUB.HQ.

APPENDIX C ₵ JOB CONTROL WORDS

Several Job Control Words (JCWs) are utilized by the FLEXIBASE software. The following is a list of the JCWs and their functions:

FLEXTERM

This JCW controls the usage of HP Screen Enhancements (e.g. inverse video etc., via the use of escape sequences), and can have 3 values:

FLEXTERM = 0   is the setting if the FLEXIBASE menu is run from an interactive terminal whose logical device either has a terminal type other than 10, 11 or 12 or does not return a value to a status request. FLEXIBASE then considers this terminal to be non-HP and thus does not use HP Screen Enhancements. This condition is automatically detected by the FLEXIBASE menu and does not need the user to set anything. Further options recognize this value as a non-HP terminal.

FLEXTERM = 1   is the setting if the FLEXIBASE menu is run from an interactive terminal whose logical device is a terminal type of 10, 11 or 12 and which returns a value to a status request. FLEXIBASE then considers this terminal to be an HP or HP-Compatible Screen and thus does use HP Screen Enhancements. This condition is automatically detected by the FLEXIBASE menu and does not need the user to set anything. Further options recognize this value as an HP terminal.

FLEXTERM = 2   if the user wishes to force FLEXIBASE to not use HP Screen Enhancements and thus override FLEXIBASE's automatic setting he/she can :SETJCW FLEXTERM=2, before running the FLEXIBASE menu. Further options recognize this value as a non-HP terminal.

FLEXTIME

This JCW controls the time delay between successive progress messages. See Appendix J for further details on this JCW. Those functions which report progress (on STDLIST and in the file FXPROGRS), do so by default at intervals of 30 seconds. This time interval can be changed as the value of FLEXTIME.

E.g. If the user inputs

```
:SETJCW FLEXTIME=10
```

before the run, then progress will be reported every 10 seconds.

Obviously more frequent progress reporting causes a slightly bigger overhead of run time.

FLEXMODE

This is automatically set by FLEXIBASE and is used internally to indicate whether the software is running in an interactive session or in a batch non-interactive job.

FLEXMODE > 0 indicates the software is running in a session.
FLEXMODE <= 0 indicates the software is running in a job.

FLEXAUTO

This is automatically set to a value of 1 in the automatic capacity management function **ONLY** if one or more sets actually have capacity changes actioned, else it is not set. This can enable the user to take

different actions in a jobstream (e.g. DELETING STDLIST or INFORMING USER) depending on whether or not capacity changes have taken place.

FXCAPJCW
If a value is set in this JCW, it can be used in the parameters controlling automatic capacity management by using "JCW" in the capacity change formula. See the Section entitled "Automatic Capacity Function" for more information.

FLEXTAPE
This JCW can be used to control operator intervention on capacity changes and repacks when disc space is low. See the Sections entitled "Capacity Function" and "Repack Function" for more information.

FXDIAG02
If this JCW is set to 1, the DIAGNOSE function will not recommend a new capacity for any set (see Section 3.5 for more information).

FXANAL01
If this JCW is set to 1, the rules normally imposed by ANALYZE that require all log files in a sequence to be present are relaxed. See Section 5.10 for further information.

FXBIGSET
If this JCW is set to 1, then the I/O optimisation is changed in the TRANSMUTE, REPACK and MASTER CAP CHANGE modules.This JCW should only be set to 1 where users of XL systems are processing large sets in excess of twice main memory size. In cases of very large sets, throughput can be increased by up to 25% by use of the JCW, but indiscriminate use will slow processing of smaller sets by a similar amount.

SETFULL
This JCW is set to "1" if any set is filled to its capacity during a CREATE run.

FXBESTBF
If this JCW is set to 1 before a TRANSMUTE (SCHEMA METHOD) run, then the blocking factor of all sets will be optimized during the run. Individual sets can be omitted from this optimization by use of a null blocking factor in the schema.

FXRESULT
This JCW is used in FLEXIBASE's Job Stream controls. When a job file is created via FLEXIBASE a line is automatically created to set the value of this JCW equal to one (i.e. !SETJCW FXRESULT=1). If a FLEXIBASE module completes correctly it will, at the end of the run, reset the value of this JCW to zero. Therefore, if the value of this JCW is not zero after running one of the FLEXIBASE modules in a job an error will have occurred (although this error may not necessarily be terminal). The job files will be created with control lines at the end to examine the contents of this JCW, and if greater than zero to send Console Messages to the Operator informing him/her of the abnormal termination.

Note that this JCW is only set to the initial value one by an MPE Command in a job stream. When running interactively the JCW will either be missing, or equal to zero. It was not designed for use in session, but the user may take advantage of it's presence if required. Whether running in job or session a successfully completed module will always terminate with FXRESULT set to zero.

The possible values of FXRESULT after running FLEXIBASE are as follows:
- Run completed successfully.
- A problem was encountered, but the data base is still intact.
- A problem was encountered; the data base is corrupt and should be restored.
- FLEXIBASE was unable to open the data base; the data base is unchanged.

Note that if FLEXIBASE is running against a database enabled for indexing with Omnidex or SUPERDEX, it may need to rebuild the TPI indexes (for example in the case of a REPACK). Should a

problem occur such that the database is intact and coherent, but the associated indexes are damaged, then FXRESULT will be set to one.

FXPAKDEC

During a TRANSMUTE run, by default, new or converted values are stored in P type data items in SIGNED format. (Positive signed values are stored by COBOL in a subtly different bit pattern from unsigned values). If you wish P type data items to be stored in unsigned format then "`SETJCW FXPAKDEC=0`" before commencing the TRANSMUTE run.

APPENDIX D C RECOVERING FAULTY DATA

It can occur during runs of FLEXIBASE that entries cannot be loaded into a Set. For example, if TRANSMUTEing a Set so that duplicate Search Item values are created or LOADing a Detail Set from an MPE file where chain heads are missing. FLEXIBASE will attempt to salvage these unloadable entries into a new MPE file so that they can be loaded into the database at a later time (after, for example, the missing chain heads had been added through QUERY etc.).

For example, let us assume that a Detail Dataset has been corrupted so that a broken chain condition has been detected using the DIAGNOSE module and a REPAIR is recommended. If, in addition to the broken chains, a missing chain head condition has been created (perhaps by corruption of the Search item, or deletion through IMAGE) then when running the REPAIR option it will not be possible to repair these faulty entries without correcting the missing chain head situation. FLEXIBASE handles this situation by storing the faulty entries in a binary MPE file in such a format that they can be LOADed back in using the LOAD from MPE option (option 13). This binary file can be modified by using an editor such as EDITOR if the Search Item is faulty (or alternatively by a user-written program). If chain heads are missing then they can be added through QUERY or a user program before LOADing back the entries.

This will be handled as specified above in the following options:

        17 TRANSMUTE
        18 TRANSFER
         8 REPAIR
        13 LOAD
         5 CAPACITY-CHANGE
         6 AUTOMATIC CAPACITY MANAGEMENT

The software will tell the user the name of the MPE file, which takes the format of "XP" + the first 4 characters of the Database name + set number (or in TRANSMUTE, the filename is "XPDUPE" + set number). For example, for set number 5 of the SALES database, the file created by FLEXIBASE will be called XPSALE05. If these MPE files become full (they are built with a maximum of 1024 entries) then the following entries will be lost. If the files cannot be built then the entries will also be lost.

The messages produced by FLEXIBASE to tell the user of these occurrences are as follows:

<Error causing failure>, entry stored in <MPE filename>
        This message means that FLEXIBASE has been unable to add the new entry for the reason given,
        and has therefore stored the entry in the named MPE file.

<Error causing failure><MPE filename> full or unavailable, data has been lost
        This message means that FLEXIBASE has either failed to build the new MPE file (perhaps
        because it already exists) or the file is full. The entry cannot therefore be salvaged and is
        consequently lost.

APPENDIX E C ON-LINE HELP FACILITY

At any prompt throughout the FLEXIBASE software the user has the option to enter a question mark (?) and receive help on the current prompt. After supplying the requested information the software will re-prompt the user for a response and continue as normal.

In many cases this context sensitive help will be similar in content to the information contained in this manual. However, throughout the software there are prompts for database names, dataset names or data item names. In these cases the help facility will display a list of potential responses.

For example, when prompted for a database name the help facility will give a list of all databases residing in the logon group, e.g.:

> The databases in GROUP.ACCOUNT are
>
> BASE1   BASE2   ... etc.

For dataset names the help facility will give a list of datasets within the specified database, along with each sets capacity, number of entries and percentage full, e.g.:

> Dataset Name   Type   Cap.   Entries   %Full
>
> SETONE   M   300   100   33

This also enables the user to get immediate information on the set loading situation. Simply enter the DIAGNOSE option from the main menu, select a database and at the dataset prompt enter "?" for the above details.

Similar details are available for data items in the TRANSPOSE and CREATE functions.

Sometimes the help facility refers the user to this manual for further details, in cases where a more detailed discussion is required.

APPENDIX F C OPTIMUM MASTER SET CAPACITIES

IMAGE locates Master Set entries into a Set by applying an algorithm to the Key Value of the entry which is to be located. This algorithm is designed so as to minimize the number of times that two different Key Values will produce the same location in the Set. It is not possible, however, to fully eliminate this occurrence. Whenever this happens, IMAGE will locate the secondary entry into a nearby empty position within the Set. In certain circumstances this arrangement can contribute to poor performance.

A few secondaries are, in themselves, no problem. A large number of secondaries, however, can badly affect a database's performance when adding and deleting entries. If a set is exhibiting this behavior, one must be able to accurately calculate the correct parameters to change so as to reduce this bad behavior to an acceptable level.

THE NUMBER OF SECONDARIES

This Appendix explains how the actual number of secondaries, which contribute to poor performance, can be mathematically calculated. There exists a formula for calculating the theoretical number of secondaries. This formula is:

$$S = 1 - (1 - e^{-F})) F$$

where, S = Number of Secondaries #DIV Number of Entries
F = Number of Entries #DIV Capacity of the Set
and e = Base of Natural Logarithms (2.7183)

This theoretical formula states that the Number of Secondaries only depends upon the Capacity of the Set and the Number of Entries within the Set. In non-mathematical terms what this means is that the quantity of entries which can contribute to poor performance increases, in an exponential fashion, as the set gets fuller.

PRIME NUMBERS, GOOD OR BAD?

It has previously been suggested, that capacities which are Prime Numbers tend to give better results. Extensive research has shown this to be incorrect, although we can avoid the worst situations by using prime numbers (but we are also unlikely to pick the best).

However, there is a class of capacities which should definitely be avoided at all costs. Research has shown that any capacity which is a multiple of 64 should be avoided. This would appear, from the evidence, to be a general result for all Character Keys (X, P, U and Z types) to a greater or lesser degree. Numeric Keys (I, J, K and R types) also exhibit this problem but tend to be even worse.

EXTENDED THEORY

What can be said about the existing theoretical secondaries formula? The number of secondaries is obviously a function not only of the capacity and the number of entries, but also of the characteristics of the particular key value distribution. That is, for any capacity and number of entries there could be a different range of key values, and the number of secondaries will vary with this different distribution. But by how much? Well, the majority of cases seem to match the theoretical formula fairly well. However, certain cases, typically numeric or 2-word keys with poor distributions of key values, give results which on average parallel the theoretical curve, but with a constant offset (either higher, or even lower in some cases). This leads to the first mathematical result:

The number of secondaries, averaged about each capacity, tends towards the formula:

```
S = 1 - (1 - e  ) ) F + k
```

where k is the "Key Distribution Offset."

The "Key Distribution Offset" is the constant offset mentioned above, and represents an improvement, or degradation, to the theoretical quantity of secondaries due to the specific spread of key values actually in the set. Positive values of k indicate that the key values in the set tend to cluster together more. Negative values indicate a better spread of values. In many cases the value of k will be zero, but can be as high as 11%. The above formula can be used to estimate the average number of secondaries at any capacity, but only by examining the current state of affairs to measure k. More usefully one can calculate the approximate capacity which would yield a manageable number of secondaries.

It is important to note that this does not guarantee that the capacity so calculated will definitely yield this exact number of secondaries. Because of the erratic behavior of any single capacity one must perform a sampling test for a range of capacities about the chosen capacity to select the capacity which gives the lowest number of secondaries.

THE EFFECT OF BLOCKING FACTORS

When attempting to locate a secondary, IMAGE will first try the same block of entries in which the primary entry is located. Now, if a secondary is successfully located in the same block as the primary this does not lead, in itself, to a large degradation of performance. When adding, deleting or locating the secondary, IMAGE must do a little more Ȼ two attempts instead of one Ȼ but this is ideally all done in memory without any need for swapping of buffers or extra disc I/Os (if you're short on memory though, this will not help!). Measurements indicate that the time taken to access a secondary within the same block as its primary is approximately 15% to 20% of the time taken to read that block in from disc (although this will vary with processor speed Ȼ these figures are for a mid-range classic 3000 machine). In terms of elapsed time, reading a secondary within the same block as the primary takes about 1.2 times the time to read in the primary itself. Taking disc caching or XL mapped file access into account this figure actually rises to about 50% (because the time to originally read the primary decreases with caching enabled).

Three things, however, really do hurt performance:
- Large numbers of secondaries from the same primary location (i.e., long secondary chain),
- Lots of full blocks (high clustering), and
- Large numbers of secondaries in blocks other than the primary location's block (inefficient secondaries).

INEFFICIENT SECONDARIES

The term "Percentage Inefficient Secondaries" is an important statistic for an IMAGE Master Set, and is defined as the number of secondaries which lie outside of their primary block, as a percentage of the number of entries. The fact that another IMAGE Block must be read in from disc, or memory, to locate the secondary entry results in a larger overhead than that previously discussed. With or without caching the time taken to read in an inefficient secondary would be a minimum of twice the time taken to read a primary (assuming the primary was in cache to start with). If the primary was not in cache the situation improves because the inefficient secondary will be Ȼ leading to a factor of 1.5 rather than 2. One or two of these inefficient secondaries is not going to upset the apple cart very much, but when the percentage of inefficient secondaries reaches approximately 23% or more one can expect to see performance begin to degrade Ȼ not only on adding and deleting but also on calculated master reads.

To see how this value varied, a large number of experiments to load up master sets at various capacities and blocking factors, key types, number of entries etc, were performed.. The results were recorded and analyzed, and an empirical mathematical formula deduced which accurately matched the data. The formula is:

$$I = e^x$$
where $x = \ln(S) - V(B-1)$

where, S = No of Secondaries #DIV No of Entries,
   B = Blocking Factor,
   I = Number of Inefficient Secondaries,
and   V = "Key Value Randomity" (see below).

Again, in non-mathematical terms, this says that the quantity of inefficiently located entries in a set decreases, exponentially, as the blocking factor increases. The rate of decrease is determined by the factor we have called the "Key Value Randomity."

We have given this factor the name of "Key Value Randomity" because it depends upon the actual spread of key values as well as the distribution of those keys throughout the set due to IMAGE's hashing algorithm. The purpose of the hashing algorithm is to enable a random distribution of primary positions. The parameter V measures just how successful IMAGE has been in this endeavor. Typically, values of V seem to range from 0.001 (very poor randomity) to 0.700 (very good randomity).

OPTIMUM CAPACITY SELECTION PROCEDURES

The above empirical theory gives us a good basis upon which to calculate the optimum capacity for any master dataset. Some general rules are:
- Avoid Capacities which are multiples of 64.
- Powers of two are always awful, and don't select a Capacity of 1,024 under any circumstances!
- For Numeric keys avoid Capacities which are even numbers.
- In many cases only odd Capacities gave reduced Secondaries for Numeric Key Types (I, J, K & R). This is not, however, a hard and fast rule as it depends entirely on the range of Key Values within the set. By selecting a Capacity above the highest Key Value in the set it is even possible to have absolutely zero Secondaries. So, this is simply a guideline.
- Do not select a Capacity just because it's a Prime Number.
- Test the actual key values against any Capacity selected.
- Experiment shows that varying the capacity up and down by 4 or 5 will enable a capacity to be selected which results in less secondaries than any of the others. The individual variability of specific capacities must be examined before selecting the capacity, so as to weed out the bad performers.
- Take the Blocking Factor into account.
- The most important statistic of a Master Dataset is the percentage of inefficient secondaries, not the percentage of secondaries themselves. It is this figure which quantifies how inefficient in terms of performance these secondaries are C and this figure, in turn, depends upon the blocking factor.

The method of "Good Capacity Selection" first requires the current values of the two key distribution parameters k and V to be determined. In doing this we are establishing how the current spread of key values within the set affect the number of inefficient secondaries generated by IMAGE. From these two values the approximate location of a capacity which will give, say, a percentage inefficiency of 15% or less can be established (given the current blocking factor). An examination of the capacities either side of this capacity against the actual key values will lead to a capacity which will, ideally, give the smallest number of secondaries. (In the real world absolutes are unobtainable, but we can get close!)

A better alternative would be to increase the blocking factor (either as well as, or instead of, increasing the capacity). An appropriate blocking factor may also be estimated from the above empirical mathematical formulae. However, care should be taken on MPE/V systems, since too high a Blockmax reduces the number of IMAGE buffers available. Current thoughts indicate that in most cases the Blockmax should not be raised above 1024 or 1536 words (due to the potential restriction this places upon the number of buffers available). These restrictions no longer apply on MPE/iX systems, where the number of buffers is either fixed irrespective of block sizes, or made redundant (from MPE/iX 3.0 onwards) by the fact that IMAGE now uses mapped file I/O. In this case, there is no restriction on block sizes; indeed I/O efficiency can be improved by increasing the Blockmax to the maximum allowed by IMAGE.

HOW MANY SECONDARIES ARE TOO MANY?

All the above theory may be academic interest, but what practical use does it really have? Well, a reduced amount of secondaries will definitely improve DBPUT and DBDELETE performance. If an application uses DBPUTs and DBDELETEs intensively this can become crucial to application run times. Also, excessively long secondary chains can affect DBGET and DBPUT performance if IMAGE has to traverse

these chains each time a mode 7 DBGET or DBPUT is required. If a significant proportion of these secondaries lie outside of the primary's block (i.e., inefficient secondaries) then performance is further degraded.

As an illustration of how these factors actually affect performance, compare the following run times. The following timings were done on a range of datasets, varying the number of secondaries and inefficient secondaries in each set. Three operations were timed on a mid-range classic 3000 machine, single user in batch mode, with disc caching (similar results were obtained without caching). These operations were:

- a complete load of the set using DBPUTs.
- a serial DBGET & DBDELETE of the entire set.
- a serial DBGET of the entire set.

By subtracting the serial DBGET time from the DBGET/DBDELETE time we get the time taken to solely perform the DBDELETEs. Adding this to the DBPUT time and dividing by the number of entries in the particular set we get the average time taken to perform one DBPUT and one DBDELETE.

%age Blocking		Secondaries		Elapsed Seconds (per Entry)			CPU Seconds (per Entry)		
Full	Factor	%age	Ineff	PUT	DELETE	Total	PUT	DELETE	Total
58%	11	24.3%	1.4%	0.048	0.045	0.093	0.033	0.031	0.064
58%	10	24.3%	4.2%	0.048	0.046	0.094	0.033	0.031	0.064
67%	12	43.0%	9.8%	0.049	0.067	0.096	0.035	0.032	0.067
58%	2	24.3%	14.0%	0.055	0.048	0.103	0.039	0.032	0.071
80%	3	31.7%	19.1%	0.062	0.052	0.113	0.042	0.031	0.073
64%	1	25.9%	25.9%	0.092	0.066	0.158	0.048	0.029	0.077
99%	4	37.0%	30.2%	0.184	0.054	0.238	0.164	0.034	0.198
99%	2	37.0%	34.2%	0.323	0.057	0.38	0.299	0.034	0.333

These are just some of an extensive range of timings, from which one can accurately state just how many Inefficient Secondaries are too many. The difference in average timings highlights the excessive run times beyond approximately 23% Inefficient Secondaries (the "knee in the curve" occurs at just beyond 23%). In fact performance begins to drop at around the 16% level, and really falls off at 23%. The above theory can be used to establish a correct capacity which will provide this acceptable level of inefficiency. The DIAGNOSE module of FLEXIBASE has been modifed in line with this theory so as to recommend appropriate capacities for the spread of key values actually in the set.

OTHER CONSTRAINTS

Additional constraints on the selection of a Master Set Capacity are those of providing sufficient room for expansion (i.e., not too low a capacity) and remaining within acceptable serial DBGET limits (i.e., not too high a capacity). Available disc space and time must, of course, also be taken into consideration.

Something like 10% to 15% of the Set should be available purely for new entries, otherwise one would be increasing the capacity far too often. Some Sets, of course, do not change at all, with the other extreme being sets which are continually filling up and emptying out.

Any application which serially reads a Master Set is going to have its run times determined by the capacity. IMAGE will read each potential entry in the set, whether in use or not, until it reaches the last position in the set. Thus, if one selected a capacity which gave 0% inefficient secondaries at the expense of being only 15% full then serial DBGETs would really suffer.

The other area for consideration is the average length of the secondary chains within the set. It is possible to have a small number of inefficient secondaries but for them all to be on one or two long secondary chains. In this circumstance performance is degraded when accessing those secondaries. However, this is a refinement to the basic principle and is, luckily, not a common occurrence in sets with less than 30% secondaries.

APPENDIX G C TRANSACTION LOGFILES C THEIR USES

The ANALYZE module of FLEXIBASE makes use of the Transaction Logging facility provided by IMAGE and TurboIMAGE. This facility was primarily introduced for security purposes (enabling database recovery with DBRECOV). With logging enabled for a database, MPE logs all database writes (and some other database calls as well) to a logfile. These logfile records can then be used with DBRECOV.PUB.SYS to perform roll-forward recovery from a previous DBSTORE or STORE of the database.

Recovery was the original and prime purpose of Transaction Logging. But how often is this recovery facility actually used? Can't we put these logfiles to some more immediate use? If you use BACKCHAT (another product from Open Seas) you will already know that logging can be used to mirror and distribute databases. In fact there is also much valuable information locked up in these log files, information that can be put to use each day.

One aspect that sometimes concerns users when logging is suggested is the possible performance impact. In practice you can usually turn on logging and not see any performance impact whatsoever (if in doubt, just try it for a day or two as it is very easy to turn logging on and off).

Transaction Logfiles contain a complete history of all write access to a database during the time that the logging process was running. As such they can be used to establish a dynamic picture of the actual use of a database. This picture can then be merged with the static information of the database provided by the DIAGNOSE module of FLEXIBASE. In this way a clearer view of the database can be formulated.

For example, a Logfile can tell you how many users actually open the database with write access and which sets are modified most, and how often, and at what time of the day. This is information that can never be gleaned from looking at a static database in isolation, and this information is vital in establishing the criteria for performance optimization.

Logfiles can also tell you exactly who updated a certain data item, at what time and on what logical device. Sensitive data items can be monitored and unusual data behavior can be traced to the programs which cause them. In addition, logfiles can supplement or replace existing audit trails or work trails (ASCII files listing modifications done by programs and users). The little-used DBMEMO call, along with DBBEGIN and DBEND, has the facility for entering text into a logfile. This text can be designed by the programmers so as to write information to the transaction logfiles instead of to ASCII work trails. At the end of the logging cycle (see later) the logfiles can be read and those DB text calls can be listed in sorted order.

What else can you tell about a database from its logfile? Well, firstly it must be understood that only write access is logged. This means that most of the DBGET and DBFIND access that occurs in a database goes un-noticed. However, there is a lot we can tell from the write access alone. Remember, that every DBUPDATE and DBDELETE must have been preceded by at least one DBGET. So we already can establish some of the read access to the database.

However, one logfile alone gives a rather narrow picture. To build up a picture of the use of a database one must really look over a broader period of time. Today's logfile may be excessively large because of the end-of-the-month update or archiving run. Alternatively it may be a holiday period and have little in it. Either way one needs to look for a representative period of time. Now, because logfiles take up disc or tape resources it is impractical, and unworkable, to keep logfiles active for weeks at a time. So, the ANALYZE module creates and maintains a History file for the database into which the salient points from each logfile examined can be stored.

Here are some examples of the questions that can be answered by analyzing transaction logfiles:

- How many write access users have the database open concurrently?
- Which dataset gets modified the most?
- How many unnecessary DBUPDATEs do my programs perform, and which program is the worst?
- What time of day do we get most modifications?

- Which program changes the CREDIT-LIMIT field in the CUSTOMER set of the SALES database? And has any CUSTOMER been given a CREDIT-LIMIT of more than fifty thousand pounds?
- Which program keeps changing the STOCK-LEVEL of such-and-such a STOCK-ITEM to a negative quantity?
- What changes has the terminal on logical device 21 made to the Accounts database?
- Wouldn't it be nice if I could use the logfiles as an audit trail of the work done each day?
- Does the TRANSACTIONS dataset have more updates, deletes or puts done to it because I'm worried about the effect that new sorted chain is having on performance?
- If I recover from this logfile, will I lose data because of incomplete transactions?
- Every time the user runs program XYZ and enters a transaction the program hangs up for 20 seconds. There is no locking conflict so what is it doing? How many DB calls is it making?

We once had a user who reported that someone (of course it was not them!) was modifying the CREDIT-LIMIT field of a CUSTOMER record. We checked all of the programs which we knew could access this data item (all of two) and couldn't see any possibility of it being done without the user's knowledge (it was passworded and the program was tied to a specific logical device etc.). Well, we figured that the only way to be sure was to analyze the transaction logfiles every day. Each day for an entire year we checked the log files for any modifications to the CREDIT-LIMIT field, and every single one was really done by the user themself. This satisfied the user, and the external auditors.

Another time someone phoned up and said that every Tuesday the stock levels of his computer product stocks were coming up out of balance. It so happened that on Monday nights the end-of-week report was run on these stocks and it did a few updates to keep everything in line. So, of course, everybody blamed this program immediately and set about finding the bug. No luck. The program looked perfect. We ran it on a test copy of the production database and it actually worked perfectly. Now we got suspicious and blamed the users! That made us feel better but didn't fix the bug, so we checked the logfiles for Monday nights. And lo-and-behold we found another program, run automatically after the end-of-week report, which did a tidy-up routine but failed in one essential detail Ͼ it didn't update the stock levels correctly!

Another time-saving use of logfiles is in eliminating the need for some database recovery procedures. One day a payroll program (which had just been modified without being tested properly!) crashed on a Friday afternoon. Now normally this would have been a restore from the backup tapes and recover from the logfiles (after all that's what logfiles were for wasn't it?). However, as that would have taken quite some time the computer team analyzed the logfile using the ANALYZE module of FLEXIBASE. It turned out, luckily, that only one database transaction had actually occurred before the program crashed. This entry could easily be corrected by using QUERY (making copious notes as evidence in the process Ͼ a printout from ANALYZE of the full transaction details was enough) and the payroll run restarted after it had been re-coded and tested on a copy database.

However, such things are only one aspect of Transaction Logfiles. Within them lie hidden statistical information that can greatly help improve the performance of databases and individual applications. Programs such as DIAGNOSE, or the CSL's DBLOADNG, have in the past only looked at the static condition of a database at the time of the run. This is fine, but does not look at the dynamic condition of the database during normal use.

For example, long sorted paths are not usually recommended as they can severely affect performance. However, if there are very few DBPUTs and DBDELETEs on these sorted paths, their benefits may well outweigh any slight performance degradation once or twice a day. A logfile can be used to calculate the number of DBPUTs and DBDELETEs on a particular dataset.

When estimating potential capacities for a dataset it is important to know how many entries get added each day, and at what rate. Log files can tell you this also. When deciding which logical devices to place datasets upon it is important to identify the most heavily used sets. Log files can tell you which get most write access.

ANALYZE will tell you this kind of statistical information for a particular logfile but it also goes one further. One logfile in itself may be unrepresentative. It may span the month-end update run, or Xmas, and so have a larger-than average or smaller-than-average number of records. To overcome this, ANALYZE

can store each logfile's statistics into a History file for each database and thus accumulate these statistics in a more representative way.

To pick just one of the statistics produced by ANALYZE the Number of Null Updates is directly relevant to performance. These are calls to DBUPDATE which do not actually change a single value within the dataset. It is very easy to code these up in a big loop that calculates a value and updates every entry in a dataset. However, Null Updates really accomplish nothing but they do use up I/O time. Simple modifications of the guilty programs could eliminate them.

SETTING UP A DATABASE FOR LOGGING

The procedures for starting up IMAGE transaction logging are very simple and can be performed in less than 5 minutes (no application system changes are required). Follow these steps:

- Ensure that all database users have LG capacity. Use the LISTDIR5 program (on MPE/V) or the LISTUSER command (on MPE/iX) to check this.
- Set up a logging process by use of the GETLOG command. This tells MPE which logfile to use and where to find it. For example,
   ```
   GETLOG MYLOG;LOG=MYFILE,DISC
   ```
   will define a logging process with a log identifier (LOGID) of MYLOG located on disc as a file called MYFILE.
- Build the logfile itself using the MPE Build command. For example,
   ```
   :BUILD MYFILE;CODE=LOG;DISC=1000
   ```
   will build the logfile MYFILE with 1000 records with a file code of LOG (all logfiles must have this file code).
- Use DBUTIL to tell TurboIMAGE which logid should be linked to this database. For example,
   ```
   :RUN DBUTIL.PUB.SYS
   >SET MYBASE LOGID=MYLOG
   Password: (press return)
   ```
- Enable the database for logging using DBUTIL again. For example,
   ```
   >ENABLE MYBASE FOR LOGGING
   >EXIT
   ```
- It only remains now to start the actual logging process by issuing the :LOG command. For example,
   ```
   :LOG MYLOG,START
   ```
   This tells MPE to open the logfile and log any DB calls into the file associated with the logid MYLOG (defined in step 1). This step must be done on the system console by a user with OP capability.

APPENDIX H ₵ OUTPUT FILE NAMES

When using any of the FLEXIBASE software functions, file equations set the device to be used for the unloaded entries if other media becomes necessary because of a shortage of disc space, and for the listing of the details of the run. They have default values of DEV=TAPE and DEV=LP respectively. The defaults can be overridden by using file equations that are entered before running the programs. The relevant file equations are given below:

> Defaults can be overridden for example by:
>
> :FILE FXUNLOAD;DEV=CTAPE (sets the media to cartridge tape)
> :FILE FXLIST;DEV=PRINTER4,1,1 (sends listing to a specific printer)

There is one exception ₵ the RENAME function which uses a file name for the listing of TSLIST.

<u>NOTE</u>: INPUT and OUTPUT are default file names which the PASCAL compiler associates with $STDIN and $STDLIST. Since all FLEXIBASE modules are written in Pascal, this means that you should avoid setting file equations for INPUT or OUTPUT before running FLEXIBASE, otherwise the results may be unpredictable.

APPENDIX I C COMPATIBILITY WITH LOGGING

FLEXIBASE automatically disables transaction logging and ILR for the duration of all modifying runs, with the appropriate exceptions of the TRANSPOSE and LOAD functions. Because the FLEXIBASE software can change entry positions or structures it is recommended that a new log file is started immediately before or after any FLEXIBASE run. Note that FLEXIBASE will restore any logging options in effect prior to starting its processing after the processing has completed.

APPENDIX J C PROGRESS REPORTING

In order that the user can monitor the progress of a FLEXIBASE run, the percentage complete is displayed on the users terminal during sessions and is recorded in a file named FXPROGRS for both jobs and sessions during the most time consuming parts of processing. The file FXPROGRS can be read at any time. By default, the percentage is reported every 30 seconds. The frequency of reporting can be altered by the following command before the run commences: SETJCW FLEXTIME=nn where nn is the elapsed time in seconds between updates. The percentage complete is not reported for fast functions or functions using a non-homogeneous algorithm (e.g. set REPACK).

In the case of the DIAGNOSE function, since there can be concurrent users in the same group, the progress file is named FXPROGR1, FXPROGR2, etc. depending upon the user access.

APPENDIX K C NATIVE LANGUAGE FACILITY

Users have the option for FLEXIBASE user dialog and output to be in languages other than English. This is achieved by setting a file equation before running the program (it is suggested that this be incorporated into a UDC command that also contains the program run command). The following is the form of the file equations that can be used:

```
:FILE TMMESSDA.PUB.HQ=TMMESSDD.PUB.HQ for German

:FILE TMMESSDA.PUB.HQ=TMMESSDF.PUB.HQ for French

:FILE TMMESSDA.PUB.HQ=TMMESSDE.PUB.HQ for Spanish
```

Note that only German, French and Spanish are currently supported as alternative languages.

It is possible for FLEXIBASE to operate in different languages concurrently by the use of different file equates. If no file equate is present then English is assumed.

An alternative to the above is to rename the message files (e.g. rename TMMESSDD.PUB.HQ as TMMESSDA:PUB.HQ) so as to permanently replace the English language version for all users. This procedure would need to be followed whenever a new FLEXIBASE version was installed.

When a job file is run, it will always refer back to the particular language file under which it was created.

The presence of such a file equation is detected by FLEXIBASE when creating job files. If the user has issued a file equation for TMMESSDA.PUB.HQ, FLEXIBASE will add this file equation to any job file it creates so that the job will operate with the correct message file.

APPENDIX L C USING FLEXIBASE IN A POWERHOUSE ENVIRONMENT

The procedure to use FLEXIBASE in a Powerhouse environment is as follows. If using a function which changes structure i.e. TRANSMUTE or CAP CHANGE, proceed as follows:

- Decide on what changes you want to make.
- Use QDD to change the dictionary.
- Go into QUTIL and SET SOURCE *filename* NOGO
  Note that *filename* is optional. If you do not specify a name here, the default is QUTLFRC.
- CREATE ALL
- EXIT
- A message will appear in the form "FILE *filename* (or QUTLFRC) has been created".
- RUN FLEXIBAS.PUB.HQ and take option 17 from the menu.

The TRANSMUTE function will prompt you for the Old Database Name which you know and for the New Schema filename which is whatever you specified in the SET SOURCE command, or is "QUTLFRC".

APPENDIX M **C** FLEXIBASE AND APPLICATIONS PACKAGES

Some applications packages maintain information about their databases, and in particular about database sizes, in a separate dictionary. If you want to use FLEXIBASE in conjunction with such applications to maintain dataset capacities, it is important to keep the dictionary in line with the real database.

HP's MM/3000 and FA/3000

In both these products, there are in effect two dictionaries: one is used by the Customizer, while the other controls the production databases. The object is to allow the system administrator to carry out customization tasks while the application is running, and then to transfer any changes from the customizer dictionary to the production dictionary while users are off the system. This is done by using the "Prep Dictionary" job. In MM/PM and FA, the customizer dictionary is called APPDIC.

The section on "Completing Customization" in the "Customization and Operations Manual" includes a table showing "When to run the Prep Dictionary Job". This shows that any change to an existing database requires the following steps to be carried out:
- Unload the database(s)
- Generate the database.
- Run Prep Dictionary.
- Reload the database(s).

In order to integrate FLEXIBASE into this procedure, you will need to change the DB-MODIFY-FLAG in the APPDIC database in order to skip the steps that the unload and reload steps would normally carry out. The procedure is as follows:
- Make sure that there are no other customizations awaiting transfer to the production dictionary. If there are, then they must be transferred into the production dictionary as specified in the HP manual before you begin the capacity change using FLEXIBASE.
- Make sure that there are no users accessing the system, and that it has been properly backed up.
- Use the Customizer to change the dataset capacities. Note that the Customizer will sometimes round up the new capacities with a view to improving performance or blocking factors. You should note down the capacities of the datasets which you intend to change with FLEXIBASE.
- Use the Capacity Change function in FLEXIBASE to carry out the changes on the required database(s) and dataset(s).
- Use the Decompile function in FLEXIBASE to produce a new copy of the database schema file(s).
- Change (e.g. using QUERY) the DB-MODIFY-FLAG in APPDIC from "1" to "3". This in effect tells the Customizer that the database has already been unloaded and regenerated.
- Run The Prep Dictionary job. This will transfer the new capacities from APPDIC to the production dictionary.
- The Customizer will now assume that you need to reload the database(s). To skip this step, simply change the DB MODIFY-FLAG from "4" to "0".

The capacity change is now complete, and the database(s) will be consistent with both the Customizer and and the production dictionaries.

APPENDIX N ℂ JOB STREAMS

Every FLEXIBASE menu option, except for SHOW and MPE COMMANDS, can either be run interactively then and there, or can create a streamed job file. This job file will contain all the MPE and FLEXIBASE commands necessary to duplicate the function when the job is run. The default name of this file is FLEXIJOB, built in the local group and account. The file must be an 132-byte fixed ASCII file, and is built at default with 1024 records.

When setting up a streamed job file the user will first be asked to enter the User, Group and Account Passwords (if any) of the logon user. The job itself will log on using the same User, Group and Account as the logon user and must therefore supply to MPE the required passwords.

Following this the user will be able to select the name of the job file (default=FLEXIJOB). The user may supply any valid MPE file name. If the file does not already exist FLEXIBASE will build it to the above specification. If the file does already exist the user will be asked to confirm whether this file name should be re-entered or whether the job stream commands should be appended to the existing file. By appending to an existing file a full job stream can be composed of many FLEXIBASE functions in their correct order. If replace is selected the old file is deleted and a new file with the same name is created.

After each "!JOB" line in the job file there will be a number of "!COMMENT" lines identifying the user and the date/time at which the commands were entered into the file.

FLEXIBASE will then place in the job file the correct MPE command to run the required FLEXIBASE module. Following this will be a series of inputs to the module which will represent the correct sequence of inputs entered by the user. Each of these lines will also contain, at the beginning, the actual FLEXIBASE prompts used when setting up the job file. These prompts will be contained in a pair of "<<" and ">>" characters. These characters are used by FLEXIBASE to delimit the prompts so that the user input can be isolated at run time. Although the user can edit the job file's contents, and specifically the user inputs, the "<<" and ">>" characters must remain intact for successful execution when the job is run.

At the end of the job FLEXIBASE will automatically insert lines to examine the contents of the FXRESULT JCW (see Appendix C), and issue TELLOP console messages if an error has occurred. When setting up a job file, having completed the above actions FLEXIBASE will ask the user when to stream the job. Three options are available:

- the job can be streamed immediately (in which case the job file is deleted after successful streaming)
- the job can be streamed at some future day and time (in which case the job file is deleted after successful streaming)
  or
- the job file can be saved as a permanent MPE file.

Use the last option if you wish to append further FLEXIBASE modules to this job stream, or if you wish to edit the file in some way.

APPENDIX O C INDEX