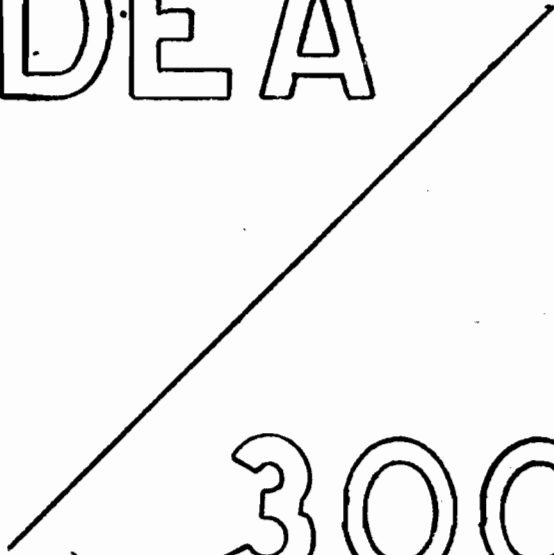


IDEA



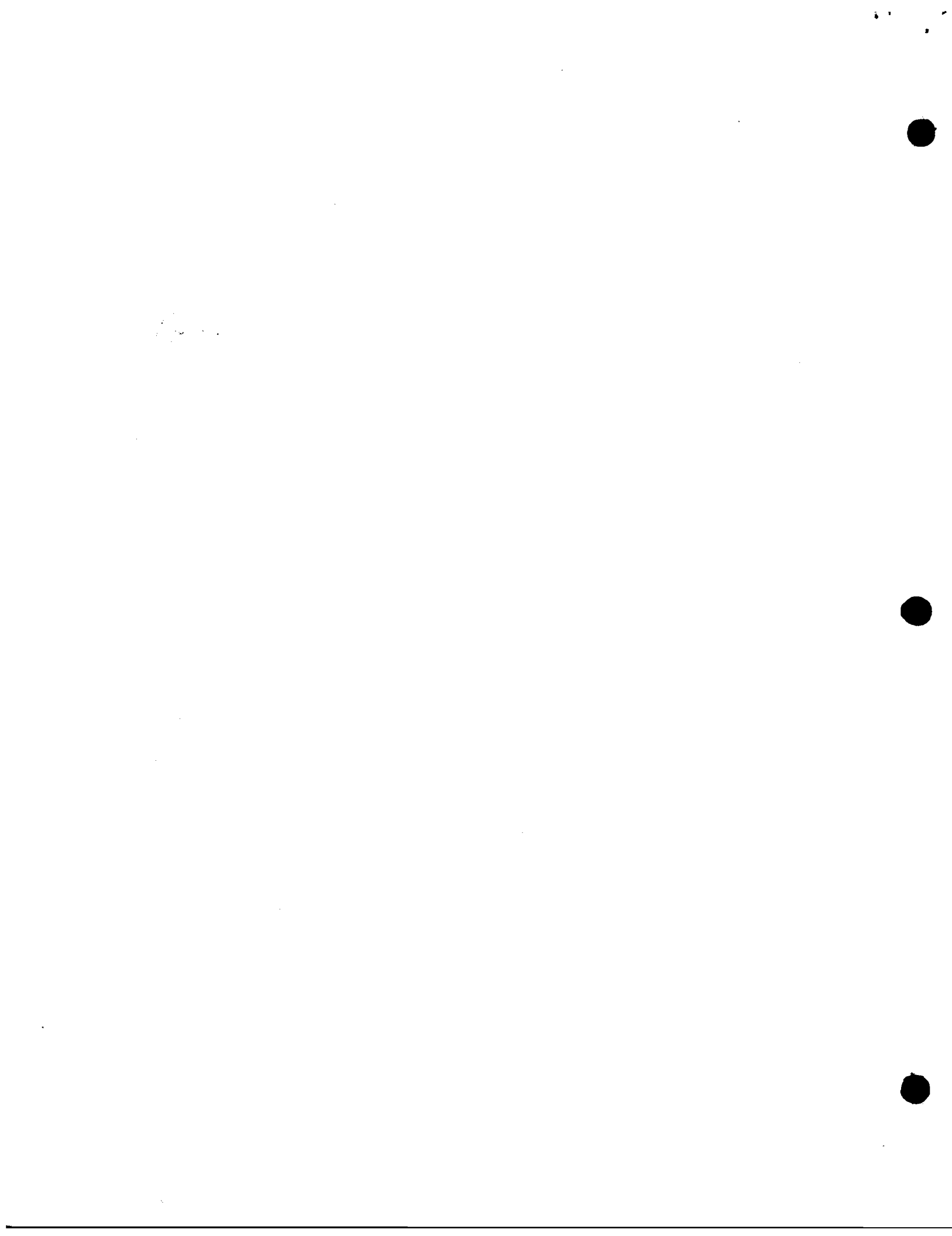
3000



REFERENCE

GUIDE

VERSION 1.1



CONTENTS	PAGE
1.0 INTRODUCTION	4
1.1 INSTALLATION	4
1.2 TEST SET-UP	4
2.0 IDEA.PUB	7
2.1 SCRIPT-FILE CREATION	7
2.2 ESTIMATE RUN	13
2.3 RUN RESULTS	16
3.0 IDEA1.PUB	16
3.1 STAND-ALONE EXECUTION	18
4.0 INTERPRETATION OF REPORTS	19
5.0 RUN CLEAN-UP	20
6.0 VALIDATION OF IDEA	21

I M P O R T A N T

It is assumed that the User of this package, IDEA/3000, has more than a casual knowledge of IMAGE/3000. Further, this package is an experimental tool and it should be executed in a non-critical environment (i.e., it should be used in a stand-alone experimental test mix; not during your production cycle). Since it is executed with multiple RIN capability, terminal lock-out or "dead-locks" can sometimes occur, which can only be relieved by a "cool" or "warm" start. Terminal lock-out will always occur if an "ABORT" is attempted during an "Estimate Run", once execution of the simulated terminal/processes has begun (i.e., let the run go to conclusion).

HP makes no express or implied warranty of any kind, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose, with regard to the program material contained herein.

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

DATA BASE DESIGN

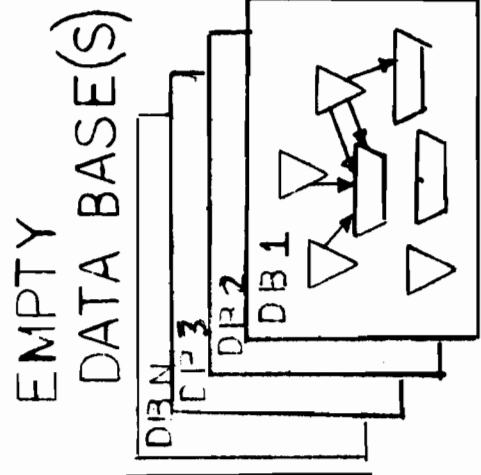
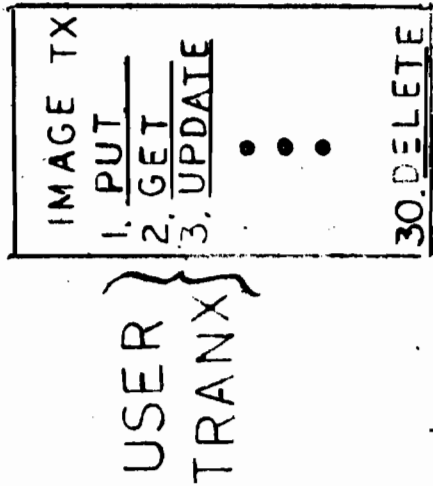
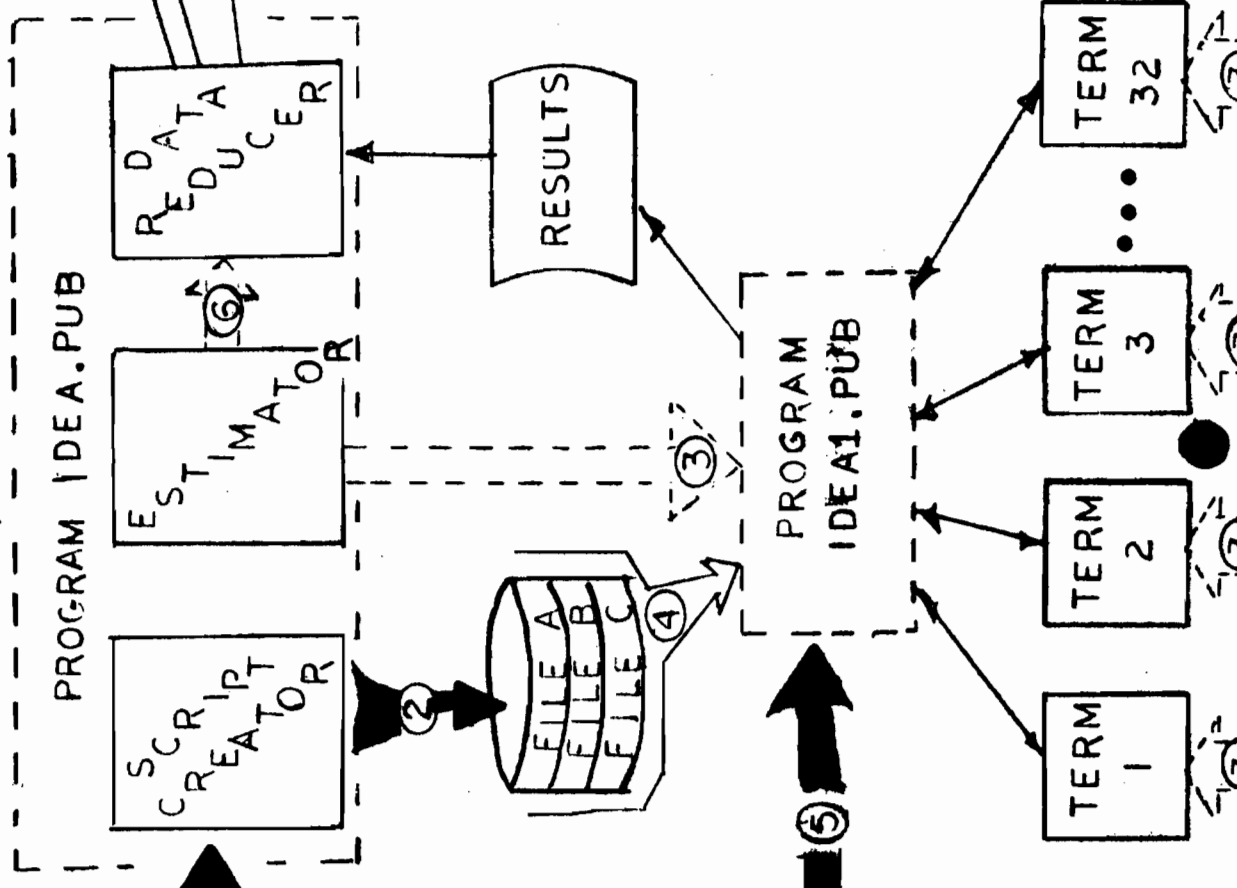


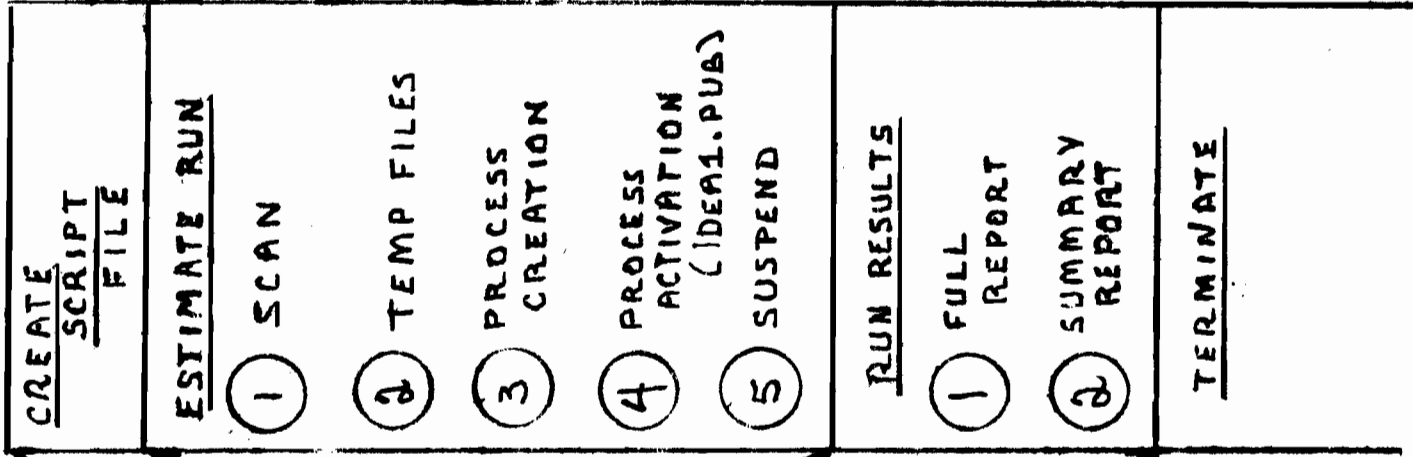
IMAGE DATA-BASE EVALUATIVE ANALYZER (IDEA/3000)



- > FAST ESTIMATE OF MIX PERFORMANCE
- INITIAL EVALUATION
- DESIGN FEEDBACK
- FUTURE CHANGE IMPACT
- > REAL ENVIRONMENT
- > SINGLE TERMINAL
- > MULTIPLE TERMINAL DISPLAY
- > SINGLE/MULTIPLE SCRIPT
- > TRUE MIX EVALUATION

FIG 1

IDEA.PUB



IDEA1.PUB

ESTIMATE RUN | STAND-ALONE EXECUTION

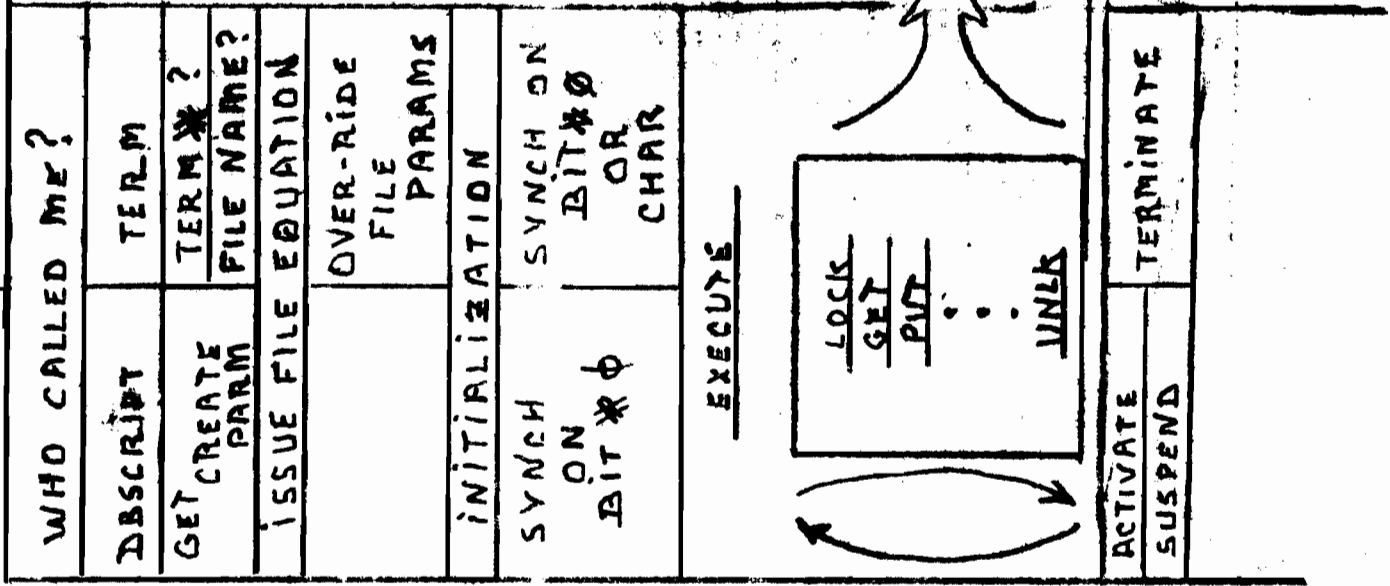


FIG 2

1.0 INTRODUCTION

The IMAGE Data-Base Evaluative Analyzer (IDEA) was designed to be a measurement tool for rapid analysis of the system performance of proposed IMAGE data base(s). The package is easily used, requiring of the User the desired data relationships in the form of an empty IMAGE data base(s) and the User's transactions in the form of "Script Files". Hopefully, this will provide the following benefits:

- a) Provide the data base designer quick assessment of the performance of his proposed data base in terms of response time and through-put.
- b) Provide the capability to the System Manager of building a model(s) of his operational data base systems, so that proposed changes to the current operational performance can be predicted without jeopardizing the "Status Quo".

1.1 INSTALLATION

The package consists of 12 files in Library format Fig. 3. It is ready to run as soon as it is "RESTORED" to the Library account. If it is necessary to recreate the program files in PUB, a "STREAM" file IDEA.JOB will do this, except for IDEA2.PUB. The source for IDEA2.PUB is IDEA5.SOURCE, but takes privileged mode capability to compile it, and do an ADDSL. Therefore, if it is necessary to work with it, someone with PM capability will have to do it. Also, PH and MR capability are necessary at the Account, Group and User level in order to use the package.

1.2 TEST SET-UP

To test the package at installation time after a "RESTORE" of the files, sign on and execute the commands shown in Fig. 4. Following this, execute the commands shown in Fig. 8. The results should be similar to those shown in Fig. 12. Stand-alone execution of IDEA1.PUB can be done following the example shown in Fig. 9.

FILES RESTORED = 12

FILE	.GROUP	.ACCOUNT	LDN	ADDRESS	<u>ORIGINAL</u>
IDEA	.SOURCE	.LIB	11	%351455	<u>NAMES</u>
IDEA1	.SOURCE	.LIB	11	%46350	<u>SCRIPTSC</u>
IDEA2	.SOURCE	.LIB	11	%413022	<u>DRIVERSC</u>
IDEA3	.SOURCE	.LIB	11	%354752	<u>FINDERSC</u>
IDEA4	.SOURCE	.LIB	11	%351544	<u>GETNUMSC</u>
IDEA5	.SOURCE	.LIB	11	%352545	<u>BITOSC</u>
IDEA	.DATA	.LIB	11	%20441	<u>DELAYSC</u>
IDEA1	.DATA	.LIB	11	%115510	<u>TIMERSC</u>
IDEA	.JOB	.LIB	11	%253762	<u>PUTTIMER</u>
IDEA	.PUB	.LIB	11	%413060	<u>STREAM</u>
IDEA1	.PUB	.LIB	11	%254141	<u>DBSCRIPT</u>
IDEA2	.PUB	.LIB	11	%413154	<u>DRIVERDB</u>
					<u>SL</u>

FILES NOT RESTORED = 0

FIG 3

:HELLO MGR.LIB,PUB
SESSION NUMBER = #S136
TUE, NOV 4, 1975, 1:40 PM
HP32000C.00.09

:RENAME IDEA7,SL
:FILE DBSTEXT=IDEA.DATA
:RUN DBSCHEMA.PUB.SYS;PARM=1

PAGE 1 HEWLETT-PACKARD 32215A.03 IMAGE/3000
TUE, NOV 4, 1975, 1:43 PM

BEGIN DATA BASE TIMER;
LEVELS:
ITEMS:
KEY,X20;
SETS:
NAME: MASTER,M;
ENTRY: KEY(0);
C: 500;
END.

DATA SET NAME	TYPE	LEVEL R W	FLD CNT	PT CT	ENTR LGTH	MED REC	CAPACITY	BLK FAC	BLK LGTH	DIS SPA
MASTER	M	0 15	1	0	10	15	500	25	377	63

TOTAL DISC SECTORS INCLUDING ROOT: 67

NUMBER OF ERROR MESSAGES: 0
HIGHEST LEVEL WORD: 0 ITEM NAME COUNT: 1 DATA SET COUNT: 1
ROOT LENGTH: 184 BUFFER LENGTH: 377 TRAILER LENGTH: 256

ROOT FILE TIMER CREATED.

END OF PROGRAM
:RUN FCOPY.PUB.SYS

HP32212A.0.03 FILE COPIER

>FROM=IDEA1.DATA;TO=PUTTIMER;NEW
EOF FOUND IN FROMFILE AFTER RECORD 2

3 RECORDS PROCESSED *** 0 ERRORS

>EXIT

END OF PROGRAM

:BUILD RESULT99;REC=54,1,F,ASCII

:RUN DBUTIL.PUB.SYS,CREATE

WHICH DATA BASE? TIMER
DATA BASE CREATED

END OF PROGRAM

2.0 IDEA.PUB

This program provides the following functions:

- a) Interactive prompting of the User to produce "Script Files" representing the environment(s) of terminals and User transactions necessary to test the proposed data base(s).
- b) Making an "Estimate Run", which analyzes, creates, and executes an appropriate sample of Script-File(s) producing a summary report of estimated timings (the "Estimate Report").
- c) Reporting function that produces a full or summary report of any timings that have already completed.

2.1 SCRIPT-FILE CREATION



A response of "1" to the prompt in Fig. 6 will initiate the creation or addition of script-records to a Script file. The Script file once created is compatible with the 3000 Editor and can be modified by it. However, when a file is kept by the Editor, only enough space is allocated to hold the current number of records. Thus, a Script file re-opened by IDEA.PUB for further additions will cause an error at close time if it has been "kept" previously by the Editor. There are several ways around this, which the experienced 3000 User is undoubtedly aware, and no time will be spent here on it.

The script shown in Fig. 6 shows the appropriate range of values and defaults with which the User can respond to the various prompts. The User should, in addition, be aware of the following conditions:

- a) "TERMINAL-NO" - This is the unique identifier of a subset (1-30) of records of the Script file. The subset represents a single User transaction (such as an Order No.) that will be executed from 1-32 terminal/processes. The next four parameters ("NO.PROCS"-through "NO.ITER'S") are associated with this group and are prompted for only on the indication of a different "TERMINAL-NO".
- b) "NO.PROCS" - This number represents the number of terminal processes that are to execute the subset of Script records associated with the "TERMINAL-NO". A maximum of 32 processes can be declared for any Script subset. However, only 32 processes can be declared for the entire Script File. From the foregoing it should be noted that a maximum of 32 "TERMINAL-NOS" could be declared of one process each with 30 associated Script records, resulting in a Script File of 960 records.
- c) "MIN/MAX PAUSE" - A random number generated between these two values will be used to delay the terminal/process at the end of every User transaction iteration representing the "think/entry-time" of someone entering data at the terminal.

TERM ID	NO. FRGS	MIN	MAX	# ITER	DATA BASE NAME	OPEN MODE	FUNC	GET TYPE	DATA-SET NAME	ITEM NAME	INITIAL-ARG VALLE
1	10	0	0	1000	TIMER	1	LOCK	--	MASTER	----	----
1	--	--	--	--	----	--	PUT	--	MASTER	KEY	ABCDEF
1	--	--	--	--	----	--	UNLK	--	MASTER	----	----

FIG 5

:HELLO MGR.LIB,DATA
SESSION NUMBER = #S3
WED, NOV 5, 1975, 10:42 AM
HP32000C.00.09

:RUN IDEA.PUB;LIB=G

SELECT: (VERSION 1.1 - MODEL 20 ONLY)

- 1 = CREATE SCRIPT-FILE
- 2 = MAKE ESTIMATE RUN
- 3 = REPORT RUN RESULTS
- CR = TERMINATE

1
SCRIPT FILE NAME IDEAL
INDICATE 1(NEW-FILE), 2(OLD-PERM) 1
RECORD NUMBER 1
01 TERMINAL-NO=1
02 NO. PROCS 10
03 MIN-PAUSE=2
04 MAX-PAUSE=9
05 NO-ITERATIONS THIS TERM-NO?=10
06 DATA-BASE NAME(REF) = ?
06 DATA-BASE NAME(REF) = TIMER
07 OPEN-MODE NO= 1
08 DATASET NAME= ?
08 DATASET NAME= MASTER
09 GET,GETU,GETD,PUT,LOCK OR UNLK?=LOCK
ANY CHGS? NO,- CR, YES, LINE-NO?=CR
MORE RECORDS FOR THIS SCRIPT-FILE(Y/CR)?Y

RECORD NUMBER 2
01 TERMINAL-NO=1
06 DATA-BASE NAME(REF) = ? TIMER
06 DATA-BASE NAME(REF) = CR
08 DATASET NAME= ? MASTER
08 DATASET NAME=CR
09 GET,GETU,GETD,PUT,LOCK OR UNLK?=PUT
11 KEY-NAME= KEY
12 INITIAL ARG (OR KEY VALUE, MAX 20 CHARS.) ?=ABCDEF
ANY CHGS? NO,- CR, YES, LINE-NO?=CR
MORE RECORDS FOR THIS SCRIPT-FILE(Y/CR)?Y

RECORD NUMBER 3
01 TERMINAL-NO=1
06 DATA-BASE NAME(REF) = ? TIMER
06 DATA-BASE NAME(REF) = CR
08 DATASET NAME= ? MASTER
08 DATASET NAME=CR
09 GET,GETU,GETD,PUT,LOCK OR UNLK?=GET
10 GET-MODE NO=2(SERIAL),4(DIRECT),5(CHAIN),7(CALCU) ?=2 (2)
ANY CHGS? NO,- CR, YES, LINE-NO?=9

09 GET,GETU,GETD,PUT,LOCK OR UNLK?=UNLK
ANY CHGS? NO,- CR, YES, LINE-NO?=CR
MORE RECORDS FOR THIS SCRIPT-FILE(Y/CR)?CR

SELECT: (VERSION 1.1 - MODEL 20 ONLY)
1 = CREATE SCRIPT-FILE
2 = MAKE ESTIMATE RUN
3 = REPORT RUN RESULTS
CR = TERMINATE

(DEFAULTS)

RANGE

1-99	(0)
1-32	(1)
0-255	(0)
0-255	(0)
0-32767	(0)
26 CHAR-MAX	
1-3	
16 CHAR-MAX	(GET)

○ OR — = User Response

16 CHAR-MAX

- d) "NO-ITERATIONS THIS TERM-NO?" - This value should represent the actual number of User type transactions per hour. If an "Estimate Run" is invoked this number will be modified to a sample size depending on the number of processes to be executed against it. The value is only meaningful when IDEA1.PUB is run stand-alone (Section 3.1). This sample size is equal to "12" for every process invoked if the "NO.PROCS" specified is 10 or less. It is equal to "8" if "NO.PROCS" is greater than 10. The sample size was determined empirically from execution of the "DEEP" program and is still being validated.

The sample size in turn determines the minimum set capacity necessary for an "Estimate Run". If:

S = sample size (12 or 8)
N = value of "NO.PROCS"
G5 = Chained Gets
G7 = Calculated Gets
P = Puts
G = all other Get types

Then:

Detail Capacity $\geq (S)(N)(P+G+G5)$
Master Cap \geq largest value of $[(S)(N)(\text{All related } G5's+G7+P+G)]$
for any Manual Master specified in the Script.

- e) The line numbers along the left side of the prompts are for reference to change the Script record before writing to the Script File.
- f) In line #9, the "U" and "D" refer to update and delete respectively.
- g) In line #10, if mode 5 (chained) is indicated, an automatic DBFIND will be executed with the chained get at execution time. It should be noted that line #9 and #10 work together to form such individual IMAGE transactions as a "chained-get-delete".
- h) If a "PUT" to a detail data set is contemplated, the Schema should be checked to see if it has any manual masters associated with it. If so, the User must make sure that these manual masters have entries with the same key value(s) as those entries that will be "PUT" into the associated detail set so that proper linkage can take place between them. Since, only a single key value can be entered for a given "PUT" Script record, all other key positions in a detail entry (as defined by the Schema) are filled with "*" (s) by IDEA.PUB. Therefore, to accomplish linkage to multiple manual masters from a given

detail entry at its "PUT" time, those other manual masters whose key is not explicitly being linked by the detail key value, will have to be linked by the "*" (s) value(s) in the detail entry. This implies that these other manual masters will have to have at least a single entry whose key value is "*" (s). If these other manual masters have been referenced explicitly (Get mode 7) or implicitly (through Get mode 5) previously in the Script, IDEA.PUB automatically places a single entry of "*" (s) in them, and they require no further attention. If not, then they, along with the manual master whose key is being explicitly referenced, must be loaded before the "PUT" to the detail set can be accomplished.

This can be done in one of two ways:

- 1) A small Script file can be created with appropriate "PUTS" to the manual masters involved and executed by IDEA1.PUB (see Stand-Alone EXECUTION, Section 3.1) prior to the execution of the Script containing the detail "PUT".

The "NO-ITERATIONS THIS TERM-NO" value entered should equal the sample-size already mentioned or the total iteration number if "Stand-Alone Execution" of the Script is planned.

The advantage to this approach is that the time to load the manual masters will not be included in the User transaction time. Its disadvantage is that a "load" Script must be executed against the data base every time before the Script-file containing the detail "PUT" can be executed.

- 2) The necessary "PUT" (s) to the manual masters can be included in the Script file ahead of the detail "PUT", but then their "PUT" time will be included in the transaction time.

The approach selected is really dependent on the User's Application being simulated.

- i) It is best to create the test data base with no security, as that will normally be the last thing to be implemented in the system application development cycle. However, the data base(s) is opened by the IDEA package with the "LEVELWORD" set to ";", in all cases.

It has been found to be easier to work with the Script creator if the User writes out his proposed script ahead of time on a form similar to the one shown in Fig. 5. Further, the position of the parameters in the Script record (Fig. 7) is critical and any modifications with the Editor should be done with great care (especially character addition/deletion). The User should make a test run with his Script with only one terminal/process, to make sure that it works properly with respect to syntax, etc.

2.2 ESTIMATE RUN

A response of "2" to the prompt in Fig. 8 will initiate an "Estimate Run". IDEA.PUB will prompt for the name of the desired Script file. The following events then take place, as IDEA.PUB scans the contents of this file. (Refer to Fig. 2).

- a) A script file* to load manual master data sets with test data is created (SCRIPT98).
- b) A script file* to load detail data sets with test data is created (SCRIPT97).
- c) A unique script file* is created for every subset of script records associated with a "TERMINAL-NO". (SCRIPT01 - SCRIPT96)
- d) Processes are then created/activated (invoking IDEA1.PUB and displaying messages to that fact) for the master script, followed by the detail script. These will load the data base(s) with appropriate test data.
- e) Processes will then be invoked for the rest of the Scripts generated from the scan done by IDEA.PUB, including any multiple processes for a given script. Messages will be issued by IDEA.PUB saying they are being activated.
- f) Each process of IDEA1.PUB then reads its script file, opens the data bases indicated in the script and when ready, displays on the initiating terminal "PROCESS XX/YY WAITING FOR S.R. BIT#0 TO BE PUT UP", where "XX" is the "TERMINAL-NO" indicated in the Script file and "YY" is the number of one (0-32) of the multiple processes indicated for that "TERMINAL-NO" in the original Script. It is the User's responsibility to put the "BIT#0" switch into the "Up" position when all processes have displayed that they are ready to proceed.
- g) Once the "Bit#0" switch has been put into the "Up" position execution will proceed until each process has finished, displaying "DRIVERDB NOW TERMINATING FOR PROCESS XX/YY", where "XX" and "YY" have the values mentioned earlier. When all the processes have finished, it is the User's responsibility to put "BIT#0" into the "Down" position.
- h) The "Estimate RUN" report will be produced (Fig. 12) automatically and the package will then terminate. For interpretation of results, refer to "4.0 Interpretation of Reports".

* Job Temporary Files

:HELLO MGR.LIB,POS
SESSION NUMBER = #S1
WED, NOV 5, 1975, 10:20 AM
HP32000C.00.09

:RUN IDEA;LIB=G

SELECT: (VERSION 1.1 - MODEL 20 ONLY)

1 = CREATE SCRIPT-FILE
2 = MAKE ESTIMATE RUN
3 = REPORT RUN RESULTS
CR = TERMINATE

2
ENTER SCRIPT-FILE NAME

PUTTIMER

PROGRAM LOADED WITH LIB = P
PROGRAM LOADED WITH LIB = P
PROGRAM LOADED WITH LIB = P
PROGRAM LOADED WITH LIB = P
PROGRAM LOADED WITH LIB = P
PROGRAM LOADED WITH LIB = P
PROGRAM LOADED WITH LIB = P
PROGRAM LOADED WITH LIB = P
PROGRAM LOADED WITH LIB = P
PROGRAM LOADED WITH LIB = P
PROGRAM LOADED WITH LIB = P
PROGRAM LOADED WITH LIB = P

O OR — = User Response

ACTIVATING MASTER-LOAD PROCESS
DRIVERDB NOW TERMINATING FOR PROCESS 98 / 0
ACTIVATING DETAIL-LOAD PROCESS
DRIVERDB NOW TERMINATING FOR PROCESS 97 / 0
ACTIVATING 01/01
ACTIVATING 01/02
ACTIVATING 01/03
ACTIVATING 01/04
ACTIVATING 01/05
ACTIVATING 01/06
ACTIVATING 01/07
ACTIVATING 01/08
ACTIVATING 01/09
ACTIVATING 01/10
PROCESS 1 / 7
WAITING FOR S.R. BIT#0 TO BE PUT UP
PROCESS 1 / 2
WAITING FOR S.R. BIT#0 TO BE PUT UP
PROCESS 1 / 1
PROCESS 1 / 4
WAITING FOR S.R. BIT#0 TO BE PUT UP
WAITING FOR S.R. BIT#0 TO BE PUT UP
PROCESS 1 / 6
PROCESS 1 / 3
PROCESS 1 / 8
PROCESS 1 / 5
PROCESS 1 / 9
WAITING FOR S.R. BIT#0 TO BE PUT UP
WAITING FOR S.R. BIT#0 TO BE PUT UP
WAITING FOR S.R. BIT#0 TO BE PUT UP
WAITING FOR S.R. BIT#0 TO BE PUT UP
WAITING FOR S.R. BIT#0 TO BE PUT UP
PROCESS 1 / 10
WAITING FOR S.R. BIT#0 TO BE PUT UP

SWITCH UP HERE

DRIVERDB NOW TERMINATING FOR PROCESS	1 /	10
DRIVERDB NOW TERMINATING FOR PROCESS	1 /	1
DRIVERDB NOW TERMINATING FOR PROCESS	1 /	6
DRIVERDB NOW TERMINATING FOR PROCESS	1 /	8
DRIVERDB NOW TERMINATING FOR PROCESS	1 /	4
DRIVERDB NOW TERMINATING FOR PROCESS	1 /	2
DRIVERDB NOW TERMINATING FOR PROCESS	1 /	5
DRIVERDB NOW TERMINATING FOR PROCESS	1 /	7
DRIVERDB NOW TERMINATING FOR PROCESS	1 /	9
DRIVERDB NOW TERMINATING FOR PROCESS	1 /	3
END OF PROGRAM		

SWITCH DOWN HERE

2.3 RUN RESULTS

A response of "3" to the select prompt will cause the prompt "FULL REPORT? (Y/N)" shown in Fig. 9. A response of "Y" will produce the full report shown in Fig. 10 along with the summary report shown in Fig. 11. A response of "N" will produce only the summary report. For a full discussion of the reports, refer to section 4.0.

3.0 IDEA1.PUB

This program (written in Fortran) is invoked either from IDEA.PUB during an "Estimate Run" or can be invoked directly from a terminal (Fig. 9). In either case the following things occur inside IDEA1.PUB

- a) The script file indicated is read in completely, with all data base(s) being opened as well as the shared output file "RESULT99".
- b) Once initialization has completed (refer to 3.1 for stand-alone initialization) execution begins on signal from the User.
- c) Each IMAGE type transaction is executed as encountered in the arrays into which the Script file was read. Upon completion of execution of each iteration through the Script file (or array values now) a timing record is written out to "RESULT99"
- d) For any IMAGE transaction requiring the "ARG" parameter (FIND, PUT, CHAINED GET) the second byte of its value ("INITARG") is incremented by "3", to give unique values to those parameters requiring them.
- e) A "chained get" IMAGE transaction always includes a DBFIND.
- f) An "update" or "delete" IMAGE transaction always includes some sort of "DBGET" with it.
- g) If an "IMAGE Error" occurs that is not catastrophic (i.e. CW=17), execution will continue for that particular Script file with a "DIRECTED GET" being substituted for the offending script record. This is done so that the particular terminal process will not drop out of the terminal mix, possibly invalidating the run.
- h) All entry values involved in the run are set to "*" except the "INITARG" values indicated in the Script file.

:RUN IDEA;LIB=G

TERM-NO ? 15
SCRIPT FILE-NAME = PUTTIMER

MINIMUM PAUSE = 2
MINIMUM PAUSE =
?1

MAXIMUM PAUSE = 9
MAXIMUM PAUSE =
?4

OR — = User Response

NUM ITERATIONS(1-32,767);0 SYNCHRONIZES TO S.R.?
NO. OF ITERATIONS = 10
NO. OF ITERATIONS =
?CR

ENTER 1 TO MONITOR TRANSACTIONS
?1

AT THIS POINT, TIMING IS READY TO BEGIN. START IT
FROM THIS TERMINAL BY HITTING CR.
?CR

TRANSACTION #	0	ON TERM #	15
THINK TIME(SECS)=	2		
TRANSACTION #	1	ON TERM #	15
THINK TIME(SECS)=	2		
TRANSACTION #	2	ON TERM #	15
THINK TIME(SECS)=	1		
TRANSACTION #	3	ON TERM #	15
THINK TIME(SECS)=	3		
TRANSACTION #	4	ON TERM #	15
THINK TIME(SECS)=	2		
TRANSACTION #	5	ON TERM #	15
THINK TIME(SECS)=	2		
TRANSACTION #	6	ON TERM #	15
THINK TIME(SECS)=	3		
TRANSACTION #	7	ON TERM #	15
THINK TIME(SECS)=	2		
TRANSACTION #	8	ON TERM #	15
THINK TIME(SECS)=	3		
TRANSACTION #	9	ON TERM #	15
THINK TIME(SECS)=	3		
TRANSACTION #	10	ON TERM #	15
THINK TIME(SECS)=	2		

DRIVERDB NOW TERMINATING FOR PROCESS 15 / 0
END OF PROGRAM

:RUN IDEA;LIB=G

SELECT: (VERSION 1.1 - MODEL 20 ONLY)
1 = CREATE SCRIPT-FILE
2 = MAKE ESTIMATE RUN
3 = REPORT RUN RESULTS
CR = TERMINATE

3
FULL REPORT? (Y/N)
Y

3.1 STAND-ALONE EXECUTION

If IDEA1.PUB is invoked directly from a terminal, the first four parameters are displayed (excluding "NO-PROCS") from the Script file indicated. This is done so that the User may override these values in the Script file for this particular run. If he does not wish to override a particular value, he merely hits carriage return.

The main reason for stand-alone execution is that the User may feel that the "sample-size" used during the "Estimation Run" is not truly representative of his data base if it is fairly large. Further, stand-alone execution allows the monitoring of transactions as they occur (refer to Fig 9), if he responds with a "1" to the prompt "Enter 1 to Monitor Transactions".

Since any execution of the data base(s) requires its loading with test data before execution of any Script files against it, the User should first run the intended Script file against the test data base(s) with a "NO.PROCS" set to one. IDEA.PUB will generate load test data for the data base(s) (SCRIPT98 & SCRIPT97) as well as splitting apart the Script file into subsets of Job temporary files. By "SAVE"ing the temporary files after the "Estimate Run" the User now has a Manual Master Data set load Script (SCRIPT98), a Detail data set load Script (SCRIPT97), and User transaction Scripts (SCRIPT01 - SCRIPT96 - where SCRIPT01 corresponds to the first unique "TERMINAL-NO", SCRIPT02 to the second, etc., etc.) Then the User would "ERASE" the data base(s), re-"BUILD" RESULT99 and run SCRIPT98 stand-alone; over-riding the "NO. OF ITERATIONS" with the desired number of test data records. Following this, he would run SCRIPT97, using the same number. The data base(s) is now loaded and the User can now invoke SCRIPT01, SCRIPT02, etc., from individual terminals in a stand-alone fashion using the same "NO. OF ITERATIONS" as in the load Scripts at each terminal. The start-up at each terminal is as follows (Refer to Fig. 9).

The message "TRANSACTION #,AA,ON TERM #BB", where "AA" is the Transaction number and "BB" the stand-alone terminal-number value, is issued for every transaction. For this reason, if the value of "NO-ITERATIONS THIS TERM-NO?" is other than 0, the message "AT THIS POINT, TIMING IS READY TO BEGIN. START IT FROM THIS TERMINAL BY HITTING CARRIAGE RETURN" is issued from IDEA1.PUB. The User can then synchronize all the terminals for this run by going around to each one and striking carriage return, once they have all been initialized. If the number of iterations is 0, the message "TERMINAL, XX, WAITING FOR S.R. BIT#0 TO BE PUT UP" is issued and execution on all terminals involved will begin with "BIT#0" in the "Up" position and cease when it is put "Down".

The results of a stand-alone run with IDEA1.PUB are obtained by invoking IDEA.PUB with the "RUN RESULTS" option already discussed.

Other than the differences noted in this section (3.1) stand-alone execution of IDEA1.PUB is the same as if it was invoked from IDEA.PUB.

4.0 INTERPRETATION OF REPORTS

All timing information from any run is written to "RESULT99", mentioned in section 1.2. Therefore, it is always possible to obtain the timing information of each User Transaction by merely invoking the "FULL REPORT" option of IDEA.PUB (section 2.3), even though an "Estimate Run" or "Summary Report" was initially obtained.

All timing records are automatically sorted by "TERMINAL-NO" (major field) and "SUB PROC-NO" (minor field) if any, whenever any reporting option is invoked.

- a) Full Report - This report (Fig.10) obtained after invoking "RUN IDEA.PUB;LIB=G/"REPORT RUN RESULTS"/"FULL REPORT (Y)"/, (section 2.3), is usually generated following a stand-alone execution of IDEA1.PUB, or if there is some question about the results displayed in the other two reports. The "FULL REPORT" displays each timing record as it occurred for each User transaction (not IMAGE transaction). Therefore, it contains the elements of the other two reports, which are merely summarizations of the "FULL REPORT".

The headings and "AVERAGES" lines (Fig.10), produced with every change in the sort fields are self-explanatory. The values in the columns from left to right are:

- 1) "TRANX-NO" - The number of the User Transaction executed, which is displayed during a stand-alone execution of IDEA1.PUB.

The "0" transaction is a dummy transaction issued for the purpose of obtaining the initial "clock" (CLOCK) value.

The "1" transaction, although displayed is not included in any calculations since it is always disproportionately larger ("TRANX-TIME") than the other transactions due to the "opening" of a data set.

- 2) "TRANX-ERROR" - This column displays the condition word of an IMAGE transaction-error occurring during a stand-alone execution of IDEA1.PUB. It is zero for an "ESTIMATE RUN".
- 3) The columns "START-CYCLE", "START-DELAY" and "START-TRANX" are internal machine clock times (CLOCK) representing minutes, seconds, and tenths of seconds in each column. Since the resolution of CHRONOS time is 0.1 sec., all calculations are accurate to ± 0.1 sec.
- 4) "RECORDING TIME" - The time necessary to write a timing record to "RESULT99".
- 5) "PAUSE" - The delay (if any) imposed upon the process to represent think time at the terminal.

- 6) "TOTAL-DELAY" - Delays in processing due to swapping, interrupts, time-quantums, "PAUSE", etc. (not "RECORDING TIME").
- 7) "TRANX-TIME" - The time it took to perform one iteration through the Script file for that process.
- 8) "CYCLE-TIME" - The sum of the "RECORDING TIME", "TOTAL-DELAY" and "TRANX-TIME".

It should be noted that the difference between two successive transactions is equal to the "CYCLE-TIME", (i.e. all the time is accounted for). The "AVERAGES" line values are not rounded.

- b) Summary Report - This report (Fig.11) is produced at the end of the normal "FULL REPORT" or can be invoked by "RUN IDEAL.PUB"/"REPORT RUN RESULTS"/"FULL REPORT (N)". It is merely the "AVERAGE" lines displayed from the "FULL REPORT" with the additional hourly through-put calculated for each terminal and the name of the Script file used by that terminal. Again, this report is usually run after a stand-alone execution of IDEAL.PUB, but where only a summary is desired.

The "AVG RESPONSE-TIME. . ." is merely the average of the values in the "CYCLE-TIME" column.

The "TOTAL HOURLY THRU-PUT. . ." is the sum of the values in the "HOURLY TRANX THRU-PUT" column.

- c) "Estimate Report" - This report (Fig.12) is produced automatically at the end of an "Estimate Run". It looks exactly like the "Summary Report" already described except that the name of the Script file appears at the top of the page.

5.0 RUN CLEAN-UP

It is necessary after a run of any type has been made to do three things:

- a) PURGE RESULT99, and then:
BUILD RESULT99;REC=54,1,F,ASCII - only after obtaining all desired reports.
- b) :RUN DBUTIL.PUB.SYS,ERASE - erase the data base.
- c) Sign off the system to clear all job temporary files before making an "Estimate Run" again.

6.0 VALIDATION OF IDEA

Only preliminary testing of the package has been completed (but continuing) as of this date (11/75).

However, so far these tests have shown the package to be accurate. It should be pointed out that IDEA1.PUB is essentially the original "DEEP" (S.E. Note #196), which over the past year has been used by a number of Users. They have reported that "DEEP" underestimated the speed of their transactions in a real environment, undoubtedly due to recording the timing records on a statistical data base in open mode 1. In addition, every terminal process had to have an extra IMAGE data segment, thus increasing the swapping problem.

This has been corrected with the shared file "RESULT99", and should allow much better prediction of the real environment.

Finally, it would be appreciated if any problems, successes, etc., with the package be communicated to:

Bob Johnson CE Group/Building 41
General Systems Division
Hewlett-Packard Company
11000 Wolfe Road
Cupertino, CA 95014

WED, NOV 5, 1975, 10:39 AM
IDEA/1.1

11MING REPORT
SCRIPT-FILE PUTTIMER

TERM# 15 TRANX-NO	TRANX-ERROR	START-CYCLE	RECORDING TIME	START-DELAY	PAUSE	TOTAL-DELAY	START-TRANX	TRANX-TIME	CYCLE-TIME
1	0	36 149 .3	.1 SECS	36 149 .4	2	2 .1 SECS	36 151 .5	.3 SECS	2 .5 SECS
2	0	36 151 .8	.1 SECS	36 151 .9	2	12 .9 SECS	37 1 4 .8	.4 SECS	13 .4 SECS
3	0	37 1 5 .2	.1 SECS	37 1 5 .3	1	1 .1 SECS	37 1 6 .4	.2 SECS	1 .4 SECS
4	0	37 1 6 .6	.0 SECS	37 1 6 .6	3	10 .9 SECS	37 1 17 .5	.1 SECS	11 .0 SECS
5	0	37 1 17 .6	.1 SECS	37 1 17 .7	2	12 .0 SECS	37 1 29 .7	.6 SECS	12 .7 SECS
6	0	37 1 30 .3	.1 SECS	37 1 30 .4	2	2 .2 SECS	37 1 32 .6	.1 SECS	2 .4 SECS
7	0	37 1 32 .7	.1 SECS	37 1 32 .8	3	10 .8 SECS	37 1 43 .6	1 .2 SECS	12 .1 SECS
8	0	37 1 44 .8	.3 SECS	37 1 45 .1	2	9 .8 SECS	37 1 54 .9	.4 SECS	10 .5 SECS
9	0	37 1 55 .3	.3 SECS	37 1 55 .6	3	3 .1 SECS	37 1 58 .7	.1 SECS	3 .5 SECS
10	0	37 1 58 .8	.1 SECS	37 1 58 .9	3	9 .0 SECS	38 1 7 .9	.4 SECS	9 .5 SECS

AVERAGES			.1 SECS		2 SECS	7 .9 SECS		.3	8 .5 SECS

FIG 10

TERM/SUB PROC-NO:	TOTAL# TRANX	RECORDING TIME	PAUSE	TOTAL-DELAY	(RESPONSE) TRANX-TIME	CYCLE-TIME	HOURLY TRANX THRU-PUT	SCRIPT-FILE
15/ 0	9	.1 SECS	2 SECS	7.9 SECS	.3 SECS	8.5 SECS	424	PUTTIMER

AVG RESPONSE-TIME FOR TERMINAL MIX

.3 SECS

TOTAL HOURLY THRU-PUT FOR THE MIX

424

FIG II

WED. NOV 5, 1975, 10:30 AM
IDEA/1.1

ESTIMATE REPORT (AVERAGES)
SCRIPT-FILE PUTTIMER

TERM/SUB PROC-NO1	TOTAL# TRANX	RECORDING TIME	PAUSE	TOTAL-DELAY	(RESPONSE) TRANX-TIME	CYCLE-TIME	HOURLY TRANX THRU-PUT
1/ 1	11	.0 SECS	4 SECS	5.1 SECS	.2 SECS	5.4 SECS	667
1/ 2	11	.2 SECS	5 SECS	6.2 SECS	.2 SECS	6.7 SECS	537
1/ 3	11	.0 SECS	6 SECS	6.7 SECS	.3 SECS	7.1 SECS	507
1/ 4	11	.0 SECS	5 SECS	5.7 SECS	.3 SECS	6.1 SECS	590
1/ 5	11	.0 SECS	5 SECS	6.2 SECS	.2 SECS	6.5 SECS	554
1/ 6	11	.0 SECS	5 SECS	5.9 SECS	.2 SECS	6.2 SECS	581
1/ 7	11	.1 SECS	5 SECS	6.1 SECS	.2 SECS	6.4 SECS	563
1/ 8	11	.0 SECS	4 SECS	5.0 SECS	.2 SECS	5.4 SECS	667
1/ 9	11	.0 SECS	6 SECS	6.3 SECS	.3 SECS	6.7 SECS	537
1/10	11	.0 SECS	4 SECS	4.6 SECS	.3 SECS	5.0 SECS	720

AVG RESPONSE-TIME FOR TERMINAL MIX

.2 SECS

TOTAL HOURLY THRU-PUT FOR THE MIX

5923

FIG 12