

Oxford University Computing Laboratory

Computer Manuals

4310

OPERATING SYSTEMS

GEORGE 3 & 4 [MK.7] Operating

3rd.Ed. 1971

1900

RECEIVED 30 NOV 1971

MANUAL (NOTICE NO.)

10/11/71

4271

GEORGE 3 AND 4 OPERATION MANAGEMENT (4)

4310 ✓

GEORGE 3 AND 4 OPERATING (1)

OXFORD UNIVERSITY COMPUTING LABORATORY
Copy 1. Library 4310
COMPUTING SERVICE

File one copy of this notice with each of the manuals indicated.

COPYING DUMP TAPES

If a user, for any reason, wishes to copy a dump tape, he should use the program #XRMM. This is particularly important with dump tapes produced under Mark 6 and it is particularly important that #XKYA should not be used.

THE ALLOCATE COMMAND: AVAILABILITY AND SUGGESTIONS AS TO USE

The ALLOCATE command will be available in Mark 6.2. It enables the installation manager to allocate filestore files to specific filestore residences in cases where he prefers to do this.

Great care must be taken not to overload a residence. If a residence is to hold directories and system files, which cannot be thrown off by the unjammer, it is imperative to allow unused space on the residence which these files will require when being updated (see point 2 under "Additions to ALLOCATE Command Specification" below). If a jam occurs on such a residence, the operator can CANCEL ALLOCATE for some of the files on the residence, but he may have difficulty deciding which residence is jammed, since at the moment there is no message output telling the operator this. (It is a planned enhancement to output a message when a residence becomes jammed.)

LISTDUMP SYSTEM MACRO IN CONJUNCTION WITH ALLOCATE

To enable the installation manager to check up on the files allocated to special residences, the LISTDUMP system macro is being released, although its main purpose is for use with the Tape to Tape Processor (TTTP) which has not yet been released. A complete specification of LISTDUMP is included in this User Notice.

The relevant files from the tape ICLKG3SYSTEM issued with Mark 6.2 are

:MACROS.LISTDUMP

and :DUMPER.PROGRAM G3DR

(These can be copied into filestore using FILEIN).

The LISTDUMP facilities relevant to ALLOCATE are concerned with details of special residences. A "Size of Special Residences" table is always output. If FULLIST is specified, an additional listing is output which shows, at the time of a given increment, which files were allocated to residences ("S" suffix in "ONLINE UNIT" column), the number of the residence they occupy at the time of running LISTDUMP ("ONLINE UNIT") and the size of the file when dumped ("SIZE" column). Thus, running

LISTDUMP

or LISTDUMP FULLIST

immediately after a dump would give an up-to-date picture of the loading of special residences.

ADDITIONS TO ALLOCATE COMMAND SPECIFICATION

- 1 Besides being able to ALLOCATE files to special residences one can also ALLOCATE to non-special residences. Note, however, that files ALLOCATED to non-special residences will not necessarily be present on that residence after a general restore. The facility should be regarded as a way of experimenting with specific files on different residences without the need to create special residences.
- 2 The installation parameter BACKJAM will not apply to special residences, but will effectively have a value of 100% for these residences. Note however that, as for all residences, 25K words will be reserved for the exclusive use of the Backing Store Unjamming system and will not, therefore, be available for ALLOCATEing. Special residences containing directories and system files should not be ALLOCATED even up to this limit, as extra blocks will be required in updating these files - up to 90% or up to 95% of this limit would probably be reasonable.
- 3 Workfiles or multifiles may not be ALLOCATED.
- 4 The abbreviation of ALLOCATE is AO.

LISTDUMP SYSTEM MACRO

Function

Lists the state of the filestore as held on a GEORGE 3 dump tape for any required increment. It can be used as a free standing program or as a preliminary to the Tape to Tape Processor.

Format

LISTDUMP *parameter list*

where entries in the parameter list are as described below.

All parameters to the command are optional and may be placed in any order.

<i>Parameter format</i>	<i>Function</i>	<i>Default action if omitted</i>
USER <i>username</i>	This causes information pertaining only to the named user and his inferiors to be listed. The program will not run in TTTP mode unless the user is :MASTER.	The user :MASTER is assumed.
INC <i>increment number</i>	This causes information from this increment to be listed. If run in TTTP mode, the TTTP files are checked to ensure no overwriting can take place.	Information from the latest increment will be listed provided that, in TTTP mode, no overwriting of earlier information occurs.
TAPE <i>magnetic tape description</i>	If this parameter is present the INC <i>increment number</i> parameter must also be present. (See <i>Execution</i> .) It is not allowed in TTTP mode.	The first suitable tape containing the required increment is used.
MODETTTP	This causes the additional output to :DUMPER.DUMPER TAPES, :DUMPER.USEFUL FILES and :DUMPER.INC LENGTHS to be given (see <i>Execution</i>)	The program is run in NOTTTP mode, and output to the above files (see <i>Function</i>) is suppressed.

<i>Parameter format</i>	<i>Function</i>	<i>Default action if omitted</i>
MODEINCLIST	Lists :SYSTEM. INCINDEX only. This facility is included mainly for the LISTINCINDEX macro, LX.	LISTDUMP is run normally in NOTTTP mode.
FULLIST or NOLIST	Defines the level of line printer output required. In FULLIST mode information on each filestore entrant and each owned magnetic tape is produced. In NOLIST mode all line printer output is suppressed. It is not allowed in TTTP mode or if FULLIST is also specified. (See <i>Execution</i>)	Line printer out- put in SHORTLIST mode will be given. This produces a line of output for each user giving the number of files and tapes etc. but not information about the individual entrants.

LISTDUMP in revision mode

By running LISTDUMP in REVISION mode, certain parameters in PROGRAM G3DR can be altered.

The command takes the form:

LISTDUMP REVISION *parameter list*

where entries in the parameter list specify the revision to be carried out. Note that REVISION must be the first parameter, but the other parameters are optional and may be placed in any order.

<i>Parameter format</i>	<i>Function</i>
LINES <i>n</i>	the number of lines/page of line printer output will be set to <i>n</i> . The initial value is 60.
DTLENGTH <i>n</i>	the length of DUMPER TAPES records will be set to <i>n</i> words. The initial value is 255.

GEORGE 3 AND 4 OPERATING MANAGEMENT (4)
GEORGE 3 AND 4 OPERATING (1)

<i>Parameter format</i>	<i>Function</i>
UFLLENGTH <i>n</i>	the length of USEFUL FILES records will be set to <i>n</i> words. The initial value is 255.
ILLENGTH <i>n</i>	the length of INC LENGTHS records will be set to <i>n</i> words. The initial value is 255.
TOPRESIDENCE <i>n</i>	the number of the highest filestore residence. The initial value is 63.
TAPES <i>n</i>	the maximum number of copies per increment will be set to <i>n</i> tapes. The initial value is 8.
DEADWOODDATE <i>n</i>	defines the age of "deadwood". A file is called deadwood if it has not been accessed for <i>n</i> days. The initial value is 28.
RE ON } RE OFF }	causes the program to (or not to) switch REALTIME ON internally as soon as it obtains a dump tape to read. Initially RE OFF is assumed.
REALTIME	this checks the current REALTIME status of the program and gives an appropriate message.

If, in the first seven cases, the number *n* is omitted then the current value is printed alongside the parameter listing.

If no parameter errors are made, the program will then overwrite itself in the filestore with the new version.

If contradicting parameters are given, then the right-most parameter is used.

Only the first four non-space characters are significant in identifying the parameter.

Where relevant, a check is made at run-time that these parameter values are reasonable and, if they are not, their initial values are used.

Forbidden contexts

USER except when under :DUMPER.

Initialisation

Before LISTDUMP may be run, the files PROGRAM G3DR and LISTDUMP must be copied from the latest ICLKG3SYSTEM system tape. FILEIN can be used for this.

If LISTDUMP is being used in conjunction with TTTP, the files, LITTLE INDEX and DUMPER TAPES must be set up (and given a WRITE trap) and after the first run the files USEFUL FILES and INC LENGTHS will also have been set up so these too must be given WRITE traps. The following job, if run before LISTDUMP is first used, will deal with these problems:

```
JOB LISTDUMPINIT,:DUMPER
IN DUMPERVTAPES(TG(ALL)),T????
@BEGVVVVVVV006/02/5223/59/59
T????
CY DUMPERVTAPES,USEFULVFILES(TG(ALL))
CY DUMPERVTAPES,INCVLENGTHS(TG(ALL))
IN LITTLEVINDEXTG(ALL),T????
????
EJ
****
```

Execution

The LISTDUMP macro loads and enters PROGRAM G3DR. The parameters are analysed and :SYSTEM.INCINDEX is read and listed if either INC *increment number* or TAPE *magnetic tape description* is not specified. The storage arrays for any of the other files required (for example, DUMPER TAPES in MODETTTP) are set up at this point. The specified increment is read and each entrant is processed in turn, the amount of information output being determined by the appropriate listing mode. The online size, special residences and offline size tables are updated for each entrant.

If TAPE *magnetic tape description* is specified, the program will attempt to ONLINE the required tape. In the event of an operator CANTDO or a read failure the program deletes itself and the job is terminated.

If TAPE *magnetic tape description* is not given, :SYSTEM.INCINDEX is searched for the increment to be listed and the program tries to find an acceptable tape holding this increment and attempts to ONLINE it. In the event of an operator CANTDOing the tape the program looks for an alternative one. This alternative search also occurs if there is a read failure on the selected tape while actually running the program unless the command was issued in USER and OFFLINE context.

DUMPER TAPES will contain a list of tsns of all tapes in :DUMPER's directory called GEO3PROCTAPE.

USEFUL FILES will contain pairs of words, the first of a pair containing an increment number and the second the file number of a "useful file" within that increment. A file is "useful" if no later copy of it exists and it has not been ERASEd.

INC LENGTHS will also contain pairs of words, this time holding an increment number and the total size of useful files in that increment. All current increments will have an entry pair in this file.

LISTDUMP provides the following listings:

- 1 a table of status and magstate (magnetic tape status) word meanings in :SYSTEM.INCINDEX
- 2 a listing of (the online copy of) :SYSTEM.INCINDEX, unless TAPE *magnetic tape description* is specified.
- 3 a table giving the size of the processed increments.
- 4 a summary of the directory entries in the file-store.
- 5 a table giving the size of the filestore on-line on each filestore residence.
- 6 a table giving the total size of files allocated to special residences.
- 7 a table giving the size of the off-line filestore.

Most of the line printer output is self explanatory so only a few parts of it are explained here.

All size columns are in units of filestore blocks (512 words).

Listing 4, the summary of directory entries is as follows:

- 1 In FULLIST mode:

Online Unit	the number of the on-line unit holding the file. An off-line file is indicated by a dash. If the file has been specially allocated to a particular residence, an 'S' is placed by the side of the position of the on-line unit number for the file.

Note: If, at the time LISTDUMP is run, the residence that a file has been allocated to is not present, the on-line unit (if any) that the file is on

will not be the same as that denoted by the 'S'.

Time between Accesses	this is the value of the number stored in the EAVACC word of the directory entry and is the weighted average time (in fifths of a second, in the current mark) between accesses when the file was last accessed.
Own Traps	'ERAW' (execute, read, append and write) indicate which traps the user has set for himself. If prefixed by "G" it indicates that the only trap the user has set himself is a group trap.
Other Traps	the number of traps set for the entrant other than those mentioned under "own traps".
Total Size	this column is only used for directory entrants and gives the size of the filestore tree inferior to the directory.

2 In SHORTLIST mode:

Level	this is the level of the user in the filestore tree (:MASTER is at level 1); it is only printed on the first line and subsequently when the level changes.
Size	the size of the user's files (on and off line).
Online Size	the on-line size of the user's files.
Total Size	the total size of the filestore tree inferior to the user.
Average "Formula"	the average filestore usage by the user for all his terminal files

$$= \frac{\sum_{i=1}^n (\text{size of user's files} \times \text{EAVACC})}{n}$$

where n is the number of user's files

Note that if either the value

of EAVACC for any one file or "formula" exceeds $2^{23} - 1$, the value of formula is replaced by the word "INFINITY".

Deadwood gives the number of files that have not been accessed for the number of days specified by DEADWOODATE.

Listing 7, the size of the off-line filestore:

Offline Size the total size of the "useful" files (see note above) in this increment which are not online.

Useful Size the total size of "useful" files in this increment.

Files accessed

<i>Entrant</i>	<i>Access modes</i>	
:MACROS.LISTDUMP	EXECUTE	
:DUMPER.PROGRAM G3DR	EXECUTE	
:DUMPER.LITTLE INDEX	READ	
:DUMPER.DUMPER TAPES	READ, WRITE	} MODETTTP only
:DUMPER.USEFUL FILES	WRITE	
:DUMPER.INC LENGTHS	WRITE	
:SYSTEM.INCINDEX	READ	

Error messages

1 Issued by the program:

increment number IMPROPERLY SPECIFIED

magnetic tape description GIVEN BUT NO *increment number*

ERROR IN *magnetic tape description* OR TAPE UNAVAILABLE

INCREMENT NOT FOUND IN INCINDEX

DUMP NOT FOUND ON THIS TAPE

USER NOT FOUND IN THIS DUMP

INC UNSUITABLE FOR GENERAL RESTORE

REALTIME CANNOT BE SWITCHED ON

TAPE NOT A DUMP TAPE

ONLY "USERMASTER" IN TTTP MODE

OVERWRITING LATER TTTP DETAILS
INCINDEX BUFFER TOO SMALL
INC ALREADY USED IN TTTP MODE
%(MODE) UNRECOGNISED

- 2 Issued by the macro:
 LISTDUMP ALREADY RUN IN %(MODE) MODE
 NO AVAILABLE DUMP TAPES
 TRANSFER FAILURE ON *MT IN LISTDUMP
 TWO LISTING MODES REQUESTED
 BOTH NOLIST AND NOTTTP REQUESTED
 ERROR IN LISTDUMP : MACRO ABANDONED

© International Computers Limited, Reading, 1971

RECEIVED 21 MAR 1972

MANUAL (NOTICE NO.)

22/12/71

4310

GEORGE 3 AND 4 OPERATING (2)

OXFORD UNIVERSITY COMPUTING LABORATORY

*Library
Copy 1*

COMPUTING SERVICE

File one copy of this
notice with each of the
manuals indicated.

WHATPER COMMAND

The following changes have been mended into WHATPER for Mark 6.2. These mends were not included in the initial issued versions of Mark 6. The headings indicate the areas in which the changes occur.

Magnetic tape information

Replace first four lines with:

Magnetic tape information consists of relevant section of the following:

TSN number WPR	{	(WORKTAPE IN USE BY: <i>User name</i>) (For a worktape in use by a job but not online to a program) (WORKTAPE) (For a worktape in any other state) (POOL TAPE) (XENOTAPE)

Storage unit information

Replace first line with:

Storage unit information is output for UDAS devices that are engaged (but not BEING USED BY SYSTEM) and consists of a list of the serial numbers of the storage units (SUSN) on the specified hardware unit:

UNIT *n* ENGAGED (HOLD SET) : SUSN *number(s)*

being the message output, the (HOLD SET) sub-message being output if applicable.

Notes

Section 4 should now read:

- 4 If the peripheral specified is a communications device (uniplexer, multiplexer, or communications processor (type 24)) the appropriate one of the following messages is output:

```
UNIT n WRONGED  
UNIT n ONLINE TO JOBNO m, username.job name  
UNIT n HAS MOP SWITCHED ON  
UNIT n HAS MOP SWITCHED OFF
```

© International Computers Limited, Reading 1971

RECEIVED 2 : MAR 1972

MANUAL (NOTICE NO.)

18/1/72

4310

4271

GEORGE 3 AND 4 OPERATING (3)

GEORGE 3 AND 4 OPERATION MANAGEMENT (9)

OXFORD UNIVERSITY COMPUTING LABORATORY

Copy 1

Library
COMPUTING SERVICE

4310

File one copy of this
notice with each of the
manuals indicated.THE TAPE-TO-TAPE PROCESSORAcknowledgment

LISTDUMP and TTTP (the Tape-to-Tape Processor) are based on software developed by the Universities of Hull and Oxford, respectively, whose assistance ICL wishes to acknowledge. Without their work, these valuable facilities would not have been available so soon.

Availability

The Tape-to-Tape Processor will be available in Mark 6.4. (LISTDUMP, the program which prepares data for TTTP, was made available with Mark 6.2 and was described in User Notice 4 to the *GEORGE 3 and 4 Operation Management* manual and User Notice 1 to the *GEORGE 3 and 4 Operating* manual.)

General description

TTTP processes dump tapes by copying increments directly from one tape to another, leaving out any files (including directories) that have been erased or re-dumped since the increment was created. Increments are therefore compressed into a smaller space on fewer tapes. Output tapes from TTTP are given the name GEO3PROCTAPE and will be used by the retrieval system wherever possible instead of GEO3DUMPTAPES, thereby providing faster retrieval of off-line files.

COMPARISON WITH THE DUMP TAPE PROCESSOR

TTTP provides a fundamentally different type of processing to that available with the built-in dump tape processor. In particular:

- 1 TFTP does not normally bring "useful files" (that is, files that are not obsolete) on-line to be dumped in a new increment. Thus filestore housekeeping activity during processing is minimized and the size of the on-line filestore is not altered.
- 2 An increment being processed is not removed from INCINDEX unless it is found to contain no useful files. Copies of the increment are produced in a more compact form on new tapes.
- 3 Up to four different sets of increment tapes can be kept. (The number is controlled by the DEPTH parameter.) If all tapes in the latest set are found to be unusable, tapes from older sets are available.
- 4 The off-line filestore is stored in a compact form on the set of most recently processed tapes.

GEO3PROCTAPES are suitable for RETRIEVES or further processing by TFTP or the dump tape processor, but they cannot be used for a general restore because they do not contain directories and vital system files. For this reason TFTP will not process increments that are less than a specified number of increments old.

Since increments are only normally erased by TFTP when they contain no useful files, INCINDEX may become rather large. To enable old increments that contain only a few useful files to be eliminated, a "re-dump" facility is provided with TFTP. The "re-dump" facility is similar to the dump tape processor in that it causes files to be brought on-line and then re-dumped in a new increment. However, TFTP does this over a longer time-scale than the dump tape processor; files are not brought on-line immediately but are RETRIEVED at the time of the next dump, and remain in the filestore until the dump after that. Unlike the dump tape processor, TFTP only brings files on-line when specifically requested to do so by means of the re-dump facility, and the user can therefore avoid backing store jams by requesting that only small increments be redumped.

TFTP is no less secure than the dump tape processor, and in one way provides greater security in that it allows processed copies of increments to be retained. The dump tape processor does not produce any new copies of an increment.

STRUCTURE OF TFTP

TFTP is a subject program run by a system macro of the form:

TFTP parameter list

A full specification of the macro and its parameters is given at the end of this notice.

The subject program consists of a permanent segment and several overlaid segments, and these are kept in two separate files, PROGRAM TTTP and OVERLAY TTTP. Since there are 20 different parameters that may be used with the TTTP macro, it is obviously important that the user should not have to include all the parameters each time the macro is issued, so default values for most of the parameters are held as constants in the permanent segment of the program. This enables the user to issue the macro with only those few parameters that are likely to vary with each run, such as which increment to start processing from (the FROM parameter) and what tapes to write processed increments to (the ONTO parameter).

The default values will not, of course, suit all users all of the time and the option is therefore provided to run TTTP in "revision" mode instead of normal processing mode. If the first parameter in the parameter list is REVISION, the subject program will not be run; instead, the subsequent parameters will be taken as the new default values and the constants in the permanent segment of the program amended accordingly.

EXAMPLES

1 TTTP FROM OLDUN, ONTO OLD

This will cause the Tape-to-Tape Processor to start processing from the oldest unprocessed increment and to output to the tape(s) it used for output in the previous run.

2 TTTP FROM 165

This will cause the Tape-to-Tape Processor to start processing from increment 165 onto new output tape(s).

3 TTTP FROM OLDEST, REDUMP 200

This will cause the Tape-to-tape Processor to start processing from the oldest increment in INCINDEX, and to mark for re-dumping all non-obsolete files in increments whose total "useful size" is less than 200 filestore blocks (100K). ("Useful size" means the total size of all non-obsolete files in the increment.) New output tape(s) will be used.

4 TTTP REVISION, DEPTH 0, DIAG TRUE, SAFETY 12

This will cause the default values of the parameters DEPTH, DIAG and SAFETY to be altered to 0, TRUE and 12 respectively.

INPUT

TTTP requires as data information on the current state of the filestore. This information is obtained by reading files prepared by the program LISTDUMP, and the first action of TTTP is normally to run LISTDUMP. These files are:

- 1 DUMPER TAPES This file contains a complete list of all tapes owned by :DUMPER that have the filename GEO3PROCTAPE and are not already in use in INCINDEX
- 2 USEFUL FILES This file contains an entry for every file in the filestore, containing the number of the increment in which the file was last dumped, and the number of the file within that increment.
- 3 INC LENGTHS This file contains the length of the currently useful filestore on each increment as a number of filestore blocks

These files are all serial files. The first and last records of each file contain the date and time written.

It is not essential to run LISTDUMP if it has already been run recently (as specified by the CINCS parameter); checks are then made that the information in the files is not out of date.

THE OPERATION OF THE TAPE-TO-TAPE PROCESSOR

1 Initialisation

Before the Tape-to-Tape Processor is used, the user must ensure that all increments recorded in :SYSTEM, INCINDEX were produced in Mark 5 or later. All increments produced in Mark 4 should have been processed by the Dump Tape Processor and the tapes containing them DUMP REUSED or DUMP RETURNed.

In order to run the Tape-to-Tape Processor (TTTP) it is necessary first to run the following jobs in order to bring into the filestore the relevant files:

- (a) JOB BRING1, :MACROS
 FILEIN TZZZZ
 ,ICLKG3SYSTEM(641)
 LISTDUMP
 TTTP
 TTTPGEN
 TTTPSETUP1
 TTTPSETUP2

TTTPSETUP3
ZZZZ
EJ

(b) JOB BRING2,:DUMPER
FILEIN TZZZZ
,ICLKG3SYSTEM(641)
G3DR,PROGRAMVG3DR
ZZZZ
CE OVERLAYVTTTP(*ED,BUCK1,KWORD10)
DISCPROG *CR,*MTICLKG3SYSTEM(641),EDLIB OVERLAYVTTTP
CDF OVERLAYVTTTP,0,0,0 } Note: Punchings
IMT ICLKG3SYSTEM,641 } begin in columns 7
IP TTTP,1,MT } and 16
F
EJ

(c) Type TTTPGEN PRIV_n,SYSTEM_x,DUMPER_y

on the operator's console. The parameters may be in any order and the last two are optional.

PRIV_n is a privilege not currently being used, which will be renamed DESTCOMM.

SYSTEM_x specifies the number, *x*, of units of time at urgency A to be given to :SYSTEM by :MANAGER.

DUMPER_y specifies the number, *y*, of units of time at urgency A to be given to :DUMPER by :SYSTEM.

The macro uses the job descriptions stored in TTTPSETUP1, TTTPSETUP2 and TTTPSETUP3 to organise the program in a form suitable for running in REVISION mode (by SAVEing the permanent part in a separate file called PROGRAM TTTP), to set up the files :DUMPER.LITTLEVINDEXT, :DUMPER.INCVLENGTHS, :DUMPER.USEFULVFILES and :DUMPER.DUMPERVTAPES and to set up and give to :DUMPER the DESTCOMM privilege. (If the named files have already been set up, their contents will be re-initialised.)

TTTP is designed to run at urgency A but, if no time at that urgency is available to it, it will run at the next lowest urgency at which time is available. If it is desired to run TTTP at urgency A, TTTPGEN will provide this, if the last two parameters above are given.

Files TTTPGEN, TTTPSETUP1, TTTPSETUP2 and TTTPSETUP3 may be erased once the TTTP system is set up satisfactorily

2 Running TTTP

The Tape-to-Tape Processor may be run as a background job, from a MOP terminal or from the operator's console. In all three cases, the user name must be :DUMPER and the job name must be PROC. If the macro is issued from the operators' console, a RUNJOB is issued with this user name and job name.

The information used by TTTP is provided by LISTDUMP [see LISTDUMP SYSTEM MACRO] via the files :DUMPER.DUMPER TAPES, :DUMPER.INC LENGTHS and :DUMPER.USEFUL FILES, and the TTTP macro calls LISTDUMP with the parameter MODETTTP in order to update these files, before entering the actual TTTP program. If TTTP is run with parameter NLD, it will not run LISTDUMP but will use its existing information about the filestore.

The TTTP program communicates with the macro by means of HALTs. Thus, if the program requires action by the macro, it will HALT; the macro detects this and does what is required and the program then resumes.

When the program has finished, it will have set up a file (called COMFILE) containing commands and this is used to return/reuse tapes which have become obsolete. If the run has been successful, the macro will OBEY this file before terminating.

A REVISION mode of running TTTP is provided, which allows the user to alter certain constants in the program. If TTTP is run in REVISION mode, the program merely checks and stores the parameter values in its permanent segment and HALTs. The permanent segment is then SAVED by the macro, as PROGRAM TTTP, and the macro terminates.

3 Input Tapes

TTTP reads :SYSTEM.INCINDEX in order to find possible input tapes for an increment it is about to process. From these, it chooses a tape which it thinks is best, according to a fixed algorithm. Basically, it attempts to find that tape which is non-failing for the increment and which contains most increments and has been produced most recently. If no non-failing tapes are available, TTTP uses tapes which the retrieval system has found to be failing. If even these are unavailable and only tapes found to be failing by TTTP itself are left, the increment is marked to be redumped and TTTP goes on to the next increment to be processed. If no tapes at all are available for an increment, (for example, operators have typed CANTDO to all of TTTP's requests for tapes), TTTP will ask if any will become available. If

the reply is 'YES', it will ask for tapes again but, if 'NO', it will give a warning that the increment will be lost if TTTP continues and ask if it is to proceed. If the answer is again 'YES', the increment is marked as obsolete and TTTP goes on to the next increment to be processed. If the answer is 'NO', TTTP will terminate, without having processed the increment.

4 Output tapes

If it is desired that TTTP should continue to use as output tapes those which it last processed onto, the parameter ONTO OLD should be included, as this is the only way in which this can be done. However, it should be noted that this is only possible if increments are processed contiguously (that is, if processing is to begin with the increment next after the last one processed in the previous run). A warning is given if no "old" tapes are available.

It is possible to request TTTP to begin processing onto specific tapes in :DUMPER's directory, by means of the 'ONTO t.s.n.' parameter. This may be repeated for up to six tapes but it should be noted that the tapes should have the name GEO3PROCTAPE and should have been in :DUMPER's directory before LISTDUMP was run.

The algorithm which TTTP uses for getting output tapes (assuming that it is not continuing on "old" tapes) is that it first looks for any tapes which were specified in the macro call. If none was specified, or if not enough were specified, it will next look for any other tapes in :DUMPER's directory called GEO3PROCTAPE which are not currently in use in :SYSTEM.INCINDEX and will try to use these. Finally, if it still does not have the required number of tapes, it will ask for pool tapes and, if these are procured, they are renamed GEO3PROCTAPE.

If TTTP is not allowed pool tapes (by "CANTDO LOAD"), it will proceed with the number of tapes it has already, providing this number is not less than the minimum number specified by the "COPMIN" parameter.

5 End conditions

Successive increments are processed until one of the following occurs:

- (a) An exception condition such as:
 - (i) End of input tape
 - (ii) End of output tape
 - (iii) Ratio not met for next input tape
(see 'RATIO' parameter)
 - (iv) Tape failure
etc.

- (b) TTTP reaches the final increment to be processed, as specified by:
 - (i) The TO parameter (if present)
 - or
 - (ii) The earlier of the last increment not on the current dump tapes and that increment which is SAFETY increments before the last increment in INCINDEX. (TTTP cannot of course process increments after the one from which LISTDUMP was run. Normally however LISTDUMP will have been run on the latest increment.)

In case (a), there are two possibilities. Recoverable conditions such as end of tape cause a suitable message to be output on the operator's console, followed by the message:

OK TO GO ON?

to which the operator can answer YES or NO.

If an irrecoverable error occurs, TTTP will terminate with a

TTTP TERMINATED

message.

In case (b) TTTP will terminate with the message

TTTP - PROCESSING COMPLETED

6 Updating INCINDEX

INCINDEX is the system file that holds all information relating to increment numbers and the tapes on which they are stored. When an increment is first dumped, a record is added to INCINDEX containing the number of tapes on which that increment is dumped, and their serial numbers. When this increment is processed by TTTP, the tape serial numbers of the newly produced tapes are inserted in the record in front of the existing serial numbers, and the number of tapes is updated accordingly. A single increment may be processed several times by TTTP, and each time the new tape serial numbers are inserted. The TTTP parameter DEPTH determines the number of sets of copies to be kept.

7 Notes on running TTTP

There are several points of importance to note:

- (a) Probably the most important is that tapes produced by TTTP are not suitable for general restore because TTTP does not copy directories nor vital system files; there is always a later version of these as they are always dumped. Because of this, TTTP is not allowed to process increments which are on the current dump tapes. A greater margin may be given by means of the parameter SAFETY, as TTTP is also not allowed to process beyond that increment which is SAFETY increments before the latest in INCINDEX.
- (b) Tapes produced by TTTP are called GEO3PROCTAPE and cannot thus be used by :DUMPER to dump on when they are finished with as 'GEO3PROCTAPE's, unless they are "returned" to the Librarian's pool. In general, tapes are "reused" by TTTP and it will be found that a supply of GEO3DUMPTAPES and GEO3PROCTAPES will be built up in :DUMPER's directory after TTTP has been run a number of times.
- (c) The parameter FROM OLDUN, which means start processing at the oldest non-obsolete, un-processed increment in INCINDEX, is only of use, of course, for increments which are being processed for the first time.
- (d) Each time an increment is processed, any old tapes for it have a count of times-processed increased by one. Any tapes for which this "depth" exceeds three are marked as obsolete in INCINDEX.
- (e) LITTLE INDEX contains a record of the "useful sizes" of increments for the last four times processed. Any length older than this is removed from LITTLE INDEX.
- (f) If TTTP has found tapes which it can RETURN or REUSE, it will sit waiting for a DUMP to be taken before issuing the commands to release the tapes. A message (Message (f) (iv)) is output informing the operators that TTTP is in this state.
- (g) TTTP has built-in error recovery procedures for tape failures, etc., and action is only required of the operators if it actually asks a question at the console; thus displays at the console are purely to give information.
- (h) Using only the ordinary mode of TTTP, :SYSTEM.INCINDEX would be liable to grow fairly large, because increments seldom become completely obsolete. In order to avoid this

and yet to preserve the original idea for TTTP (viz. that large numbers of files should not be brought on-line, which could cause a backing-store jam), the parameter REDUMP is provided. REDUMP causes increments to be marked in INCINDEX as "dump processed" and "to be redumped", and the Dumper, noticing these markings, will issue RETRIEVES for all "useful" files on increments so marked. (Such files will not actually be re-dumped until the *following* dump.)

If "REDUMP *n*" is specified, only increments of size less than *n* filestore blocks will be marked; other increments will be copied across to a GEO3PROCTAPE. Provided that *n* is kept fairly small and there are not a great number of increments having a useful size less than *n*, no backing store jam should result.

After all the "useful" files in marked increments have been re-dumped by the Dumper, these increments will have become obsolete and so will be removed.

- (i) A report on each run is made by TTTP to the line printer, containing details of which files were copied, which tapes were used, etc. It should be noted that whenever the "size" of an increment or file is mentioned, the units are "filestore blocks" (=512 words).
- (j) It will be noted that TTTP calculates a "compression ratio" of "old size" of increment to "new size". Because of the fact that directories and vital system files are removed on the first processing of an increment, the ratio will be found to be fairly large on the first processing but will drop off fairly rapidly for each successive processing. The ratio is really provided as a guide as to whether it is worth processing, so, for instance, if the ratio was found to be 1, nothing would be gained by processing as all files would be copied.

8 Messages

These are categorised as to when they occur.

(a) INITIALISATION AND PARAMETER ANALYSIS

- (i) TTTPROCESSING UNDER OPERATOR CONTROL
- (ii) TTTPROCESSING UNDER MOP CONTROL
- (iii) ERROR IN PARAMETER *xxxxxx*
- (iv) NO OUTPUT TAPES REQUIRED BY TTTP
- (v) WHICH INCREMENT TO BE PROCESSED FIRST?
- (vi) WHICH INCREMENT TO STOP PROCESSING?

GEORGE 3 AND 4 OPERATING (3)
GEORGE 3 AND 4 OPERATION MANAGEMENT (9)

- (vii) OLD SIZE *mmmm* ; NEW SIZE *nnnn*
- (viii) RATIO NOT MET FOR PROCESSING SO FAR
- (ix) RATIO NOT MET FOR NEXT INPUT TAPE
- (x) LISTDUMP MUST BE RUN IN ORDER TO PROCEED

(b) INPUT TAPES

- (i) NEW INPUT TAPE NEEDED FOR PROCESSING
- (ii) NO TAPES AVAILABLE FOR INCREMENT
- (iii) THE LIST OF POSSIBLE TAPES IS AS FOLLOWS
- (iv) WILL ANY BECOME AVAILABLE?
- (v) IF YOU GO ON INCREMENTS WILL BE LOST

(c) OUTPUT TAPES

- (i) NO OLD TAPES SUITABLE FOR PROCESSING
- (ii) *tsn* SPECIFIED BUT NOT AVAILABLE
- (iii) TTTP NEEDS *n* POOL TAPE(S)
- (iv) TTTPROCESSING WITH ONLY *n* COPIES
- (v) I HAVE ONLY *n* OUTPUT TAPES
- (vi) TOO MANY OUTPUT TAPES; LIST TRUNCATED
- (vii) 'COPIES' INCONSISTENT WITH 'ONTO OLD'
- (viii) *t.s.n.* IS STILL LOADED
- (ix) SHALL I UNLOAD IT ALSO?
- (x) OUTPUT TAPE FAIL
- (xi) AN OUTPUT TAPE IS FULL

(d) GENERAL

- (i) PLEASE ANSWER YES OR NO
- (ii) OK TO GO ON?
(following another message)

(e) PREMATURE TERMINATION

Messages (i) to (iv) are followed by message (v)

- (i) "TO" LESS THAN "FROM" IS NOT ALLOWED
- (ii) NO OUTPUT TAPES AVAILABLE FOR PROCESSING
- (iii) TTTP IN QUIET MODE SO FORCED TO STOP
- (iv) TTTP FAILED TO POSITION TAPE *tsn*
- (v) TTTP TERMINATED
- (vi) TTTP TERMINATED - INCORRECT PARAMETERS
- (vii) TTTP TERMINATED - ERRORS DETECTED
- (viii) TTTP TERMINATED - INCINDEX CORRUPTED?
- (ix) TTTP TERMINATED - COMMAND ERROR
- (x) IS A TTTP POSTMORTEM REQUIRED?

Messages (vi) to (ix) indicate that something has gone wrong. If a rerun is not successful, a bug report should be submitted, accompanied by as much relevant information as possible. A TTTP postmortem should normally be supplied in such cases.

(f) SUCCESSFUL TERMINATION

- (i) REVISED TTTP SAVED PERMANENTLY
(after run in Revision mode)
- (ii) TTTP - PROCESSING COMPLETED
- (iii) NO TAPES RELEASED
- (iv) TTTP IS AWITING A DUMP

(g) ERROR NUMBERS

Throughout the running of TTTP, there may occur the message:

ERROR NUMBER *n* DETECTED

If *n* is less than 800, TTTP will immediately halt with the message (e) (vii) above. A value of *n* which is greater than, or equal to 800, implies either an error while transcribing, or an error while updating INCINDEX. In either case, some form of recovery action is made before TTTP halts with message (e) (vii) or (e) (viii). The following descriptions may not mean a great deal to users but might be of interest.

<i>Error number</i>	<i>Meaning</i>
400	Failure while trying to read "LITTLE INDEX".
500	After transcription error - lost track of the failing input tape.
595	Required tape has not been found.
600	The file "INC LENGTHS" has been found to be empty.
800	Failure while trying to read "INCINDEX".
801	The correct number of tapes to be marked as obsolete have not been found.
802	Expected "dummy" increment in list has not been found.
803	Read past expected next increment in "INCINDEX".
900	A start sentinel was expected but has not been found.
901	A start of increment sentinel was expected but has not been found.
902	A sentinel was expected but has not been found.
903	An end sentinel was expected but has not been found.
904	The input tape has been found to be past the correct increment.

GEORGE 3 AND 4 OPERATING (3)
GEORGE 3 AND 4 OPERATION MANAGEMENT (9)

<i>Error number</i>	<i>Meaning</i>
905	An end sentinel has an incorrect format.
906	End of file found but not expected.
907	Incorrect sentinel count found at end of increment.
908	Parity failure found while reading a tape.
909	Parity failure found while writing to a tape.
910	A long block has been read.
911	The end of tape has been found while writing to a tape.
912	A qualifier block was expected but a tape mark has been found.
913	Wrong number of blocks found.
914	The start and end sentinel block counts disagree.
915	Wrong number of sentinels found.
916	Incorrect sentinel count found at end of file.
917	The tape is past the next file which is to be copied.

APPENDIX

TTTP SYSTEM MACRO

Function

Runs the Tape-to-Tape Processor (TTTP), which processes increment dump tapes by copying a series of increments directly from one tape to another.

Format

TTTP parameter list

A table of possible parameters is as follows:

<i>Parameter</i>	<i>Abbrevi- ation</i>	<i>Default if omitted</i>	<i>Formats not permitted in 'REVISION' mode</i>

Parameters which are either present or absent

NLD	NL	not NLD	-
OPS	OP	*not OPS	-
QUIET	QT	not QUIET	-
RETURN	RT	REUSE	-
REUSE	RU	REUSE	-
REVISION	RV	not REVISION	-

Parameters which are true or false.

DIAG	DI	FALSE	-
FREQUP	FQ	FALSE	-
POOLTAPES	PL	FALSE	-

Parameters which are followed by a character string or an integer.

CINCS	CI	3	-
COPIES	CS	2	-
COPMIN	CM	2	-
CORE	CO	present core size of the program	-
DEPTH	DE	1	-
FROM	FR	-	FROM <i>increment number</i>
ONTO	ON	existing GEO3PROCTAPE	ONTO OLD ONTO <i>t.s.n.</i>
RATIO	RA	2	-

* This parameter is included (or not) by the macro as and when required and so should not be used.

GEORGE 3 AND 4 OPERATING (3)
 GEORGE 3 AND 4 OPERATION MANAGEMENT (9)

<i>Parameter</i>	<i>Abbrevi- ation</i>	<i>Default if omitted</i>	<i>Formats not permitted in 'REVISION' mode</i>
REDUMP	RD	0	-
SAFETY	SA	10	-
TO	TO	see parameter TO increment number specification	

NOTES

- 1 RETURN and REUSE are mutually exclusive so only one should be included in the parameter list.
(If both are given, the last one specified will be accepted.)
- 2 Up to 17 parameters may be specified and any additional ones are ignored.
- 3 The first three parameters in the table (NLD, OPS and QUIET) are ignored in REVISION mode and no error is given if they are included.
- 4 TRUE and FALSE may be abbreviated to T and F respectively.

A more detailed specification of the parameters follows:

CINCS (CI)

Format	CINCS <i>integer</i>
Function	Specifies by how many increments the information produced by LISTDUMP may be out of date and still allow TTTP to run. If the information is found to be out of date the macro will ask if LISTDUMP is to be run.
Default	If the parameter is omitted, the default is 3.
Notes	The parameter is optional and has no fixed position.

COPIES (CS)

Format	COPIES <i>integer</i>
	or
	COPIES INSTPARA

Function Specifies the number of output tapes. The second format sets this number to the installation parameter DUMPTAPES.

Default If the parameter is omitted, the default value is 2.

Notes The parameter is optional and has no fixed position. The integer specified must be such that $1 \leq integer \leq 3$. The second format may be used in REVISION mode, to set COPIES to the current value of DUMPTAPES, but the value of COPIES will not change whenever DUMPTAPES does so.

COPMIN (CM)

Format COPMIN *integer*
or
COPMIN NONE

Function The first format specifies the minimum number of output tapes with which TTTP may proceed if it cannot get COPIES tape.

The second format specifies that TTTP should not ask for pool tapes. If it cannot get COPIES tapes without asking for pool tapes, it will output a message saying how many tapes it has got and ask whether it should proceed.

Default If the parameter is omitted, the default value is 2.

Notes The parameter is optional and has no fixed position. If the first format is used, the integer specified must be such that $1 \leq integer \leq 3$.

CORE (CO)

Format CORE *integer*

Function Specifies the amount of core given to the program, in words.

Default If the parameter is omitted, the default value is the amount of core specified in the program's request slip.

Notes The parameter is optional and has no fixed position. The integer must not contain commas. If an integer is specified that is less than the core size in the request slip, a command error occurs.

DEPTH (DE)

Format *DEPTH integer*

Function Specifies the number of extra sets of tapes to be kept per increment after processing, over and above the set just produced.

Default If the parameter is omitted, the default value is 1.

Notes The parameter is optional and has no fixed position. The integer specified must be such that $0 \leq integer \leq 3$.

DIAG (DI)

Format *DIAG TRUE*
 or
 DIAG FALSE

Function Specifies whether TTTP is to produce diagnostic printout (if set TRUE) or not (if set FALSE).

Default If the parameter is omitted, the default is FALSE.

Notes The parameter is optional and has no fixed position. Usually, it will not be necessary to run with this parameter set TRUE.

FREQUP (FQ)

Format *FREQUP TRUE*
 or
 FREQUP FALSE

Function Normally, :SYSTEM.INCINDEX is updated by TTTP when end of output tape is reached, when a new input tape is required and at the end of all processing. This will occur if the second format is used. If the first format is used, in addition to these occasions, :SYSTEM.INCINDEX will be updated after processing each non-obsolete increment.

Default If the parameter is omitted, the default is FALSE.

Notes The parameter is optional and has no fixed position.

FROM (FR)

Format FROM *increment number*
or
FROM OLDUN
or
FROM OLDEST

Function Specifies the increment at which TTTP is to start processing. If the second format is used, TTTP will start processing at the oldest increment which has not been processed by either the Dump Processor or TTTP. If the third format is used, TTTP will start processing at the oldest record in :SYSTEM.INCINDEX.

Default -

Notes The parameter is mandatory but has no fixed position. If the parameter is not given, TTTP will ask at which increment to start processing and will not continue until it is given. The first format is not valid in REVISION mode.

NLD (NL)

Format NLD

Function Specifies that TTTP is not to run LISTDUMP but is to use its existing information about the filestore.

Default If the parameter is omitted, LISTDUMP is run.

Notes The parameter is optional and has no fixed position. It is ignored in REVISION mode.

ONTO (ON)

Format ONTO *t.s.n.*
or
ONTO OLD

Function Specifies tapes which TTTP is to try to use as output tapes. The first format may be included in the parameter line any number of times (though only the first six tsns will be stored). The second format specifies that TTTP should use, as output tapes, those tapes which

it last used as output tapes. This can only be done if increments are processed contiguously.

Default If the parameter is omitted, TTTP will attempt to get tapes from :DUMPER called GE03PROCTAPE. If it cannot get enough tapes from :DUMPER and, if POOLTAPES (q.v.) is true, it will then attempt to get pool tapes.

Notes The parameter is optional and has no fixed position. It is not valid in REVISION mode.

OPS (OP)

Format OPS

Function This parameter is used by the macro to tell the program to direct program-issued displays and questions to the operator's console if TTTP is not being run from a MOP terminal.

Default The parameter will be included (or not) by the macro as and when required and so should not be used.

POOLTAPES (PL)

Format POOLTAPES TRUE
or
POOLTAPES FALSE

Function Specifies whether TTTP must ask the operators before getting output tapes from the pool (if set FALSE) or can proceed without asking (if set TRUE).

Default If the parameter is omitted, the default is FALSE.

Notes The parameter is optional and has no fixed position.

QUIET (QT)

Format QUIET

Function Specifies that if TTTP comes across a situation which it cannot handle without asking a question to the user, it should terminate, rather than wait for an answer.

Default If the parameter is omitted, the program will ask questions, if necessary.

Notes The parameter is optional and has no fixed position. It is ignored in REVISION mode.

RATIO (RA)

Format RATIO *integer*

Function Specifies the minimum "compression ratio" (that is, the ratio of old size to new size of increments within range) with which TTP may proceed without asking if it can process the increments specified.

Default If the parameter is omitted, the default value is 2.

Notes The parameter is optional and has no fixed position. The integer specified must be greater than or equal to 1.

REDUMP (RD)

Format REDUMP *integer*
or
REDUMP ALL

Function Specifies that TTP is to mark certain increments to be redumped. If the first format is used, those increments which are within range and which have a "useful size" less than *integer* filestore blocks (=512 words) will be marked to be redumped. If the second format is used, all increments within range will be marked to be redumped.

Default If the parameter is omitted, no increments will be marked to be redumped except those for which failure in transcription occurs, no non-failing tapes being available.

Notes The parameter is optional and has no fixed position.

RETURN (RT)

Format RETURN

Function Specifies that TTTP is to DUMP RETURN tapes in :SYSTEM.INCINDEX which have become obsolete.

Default If the parameter is omitted, TTTP will "reuse" obsolete tapes.

Notes The parameter is optional and has no fixed position. If both RETURN and REUSE (q.v.) are specified, the one last given will be accepted.

REUSE (RU)

Format REUSE

Function Specifies that TTTP is to DUMP REUSE tapes in :SYSTEM.INCINDEX which have become obsolete.

Default If the parameter is omitted, TTTP will still "reuse" obsolete tapes. (The parameter is necessary in the case when the default setting had been changed by REVISION (q.v.) to "return" tapes).

Notes The parameter is optional and has no fixed position. If both RETURN (q.v.) and REUSE are specified, the one last given will be accepted.

REVISION (RV)

Format REVISION

Function Specifies that the parameters following are to be used to reset the Default options. No processing is done in a 'REVISION' run.

Default If the parameter is omitted, TTTP will be run in normal mode.

Notes The parameter is optional but, if included, it must be the first parameter given. A parameter error will result if this is not the case or if it is the only valid parameter given.

SAFETY (SA)

Format SAFETY *integer*

Function Specifies the number of increments before the last in :SYSTEM.INCINDEX, beyond which TTTP is not allowed to process.

Default If the parameter is omitted, the default value is 10.

Notes The parameter is optional and has no fixed position. It should be noted that the limit is set to the numerical value of:
[the last increment in INCINDEX-SAFETY]
so that, if increment numbers are missing from near the end of INCINDEX, the actual margin of safety will be decreased.

TO

Format TO *increment number*

Function Specifies the increment at which TTTP is to stop processing.

Default If the parameter is omitted, TTTP will not process beyond the earliest of:

- (a) the last increment not on the current dump tapes
- (b) that increment which is SAFETY (q.v.) increments before the last increment in INCINDEX
- and (c) the increment from which LISTDUMP was run.

Notes The parameter is optional and has no fixed position. It is not valid in REVISION mode.

Execution

The macro first runs the LISTDUMP macro, if required, handing over to it any relevant parameters. LISTDUMP is used to update the information in the files :DUMPER.USEFUL FILES, :DUMPER.INC LENGTHS and :DUMPER.DUMPER TAPES, which is required by TTTP.

The TTTP program is then loaded and entered, the LISTDUMP parameters having been removed. Taking into account the parameters specified and using the information in the three files mentioned above, together with that in :SYSTEM.INCINDEX and :DUMPER.LITTLE INDEX,

TTTP decides which increments it is to process, which tapes are required as input tapes and which are available as output tapes. It then asks for the tapes required and, when it is satisfied that the right tapes have been on-lined, copies the "useful files" from the input tape to the output tape(s). (A "useful file" is defined as one which is the latest version of an extent terminal file.)

It may be that new input/output tapes will be required and the program requests these as necessary. If TTTP comes across a situation which it cannot deal with itself, it will ask a question at the MOP terminal or operator's console, as appropriate. When all the "useful files" have been copied, TTTP will issue a command to release those tapes in :SYSTEM.INCINDEX which it is possible to return or reuse.

TTTP may be run in REVISION mode, whereby the parameters are used to reset certain variables. This is achieved by SAVEing the non-overlaid part of the program.

Files accessed

Entrant

Modes of access

:MACROS.LISTDUMP	EXECUTE
:DUMPER.PROGRAM G3DR	EXECUTE
:DUMPER.PROGRAM TTTP	EXECUTE, WRITE
:DUMPER.OVERLAY TTTP	EXECUTE
:DUMPER.LITTLE INDEX	READ, APPEND
:SYSTEM.INCINDEX	READ, APPEND
:DUMPER.USEFUL FILES	READ, WRITE
:DUMPER.INC LENGTHS	READ, WRITE
:DUMPER.DUMPER TAPES	READ, WRITE
:DUMPER.COMFILE	EXECUTE, WRITE

Messages

For a complete list of these, see item 8 under
The Operation of the Tape-to-Tape Processor.

1. The first part of the document discusses the importance of maintaining accurate records of all transactions and the role of the accounting department in ensuring the integrity of the financial data. It emphasizes the need for transparency and accountability in all financial reporting.

2. The second part of the document outlines the various methods used to collect and analyze financial data, including the use of spreadsheets, databases, and specialized accounting software. It also discusses the importance of regular audits and the role of external auditors in verifying the accuracy of the financial statements.

3. The third part of the document focuses on the importance of budgeting and financial planning. It discusses the various factors that can affect a company's financial performance and the need for a proactive approach to managing financial resources. It also outlines the various tools and techniques used to develop and monitor a budget.

4. The fourth part of the document discusses the importance of risk management and the role of the accounting department in identifying and mitigating financial risks. It outlines the various types of risks that a company may face and the various strategies used to manage these risks.

5. The fifth part of the document discusses the importance of financial reporting and the role of the accounting department in preparing and presenting financial statements. It outlines the various types of financial statements that a company may prepare and the various standards that must be followed in preparing these statements.

6. The sixth part of the document discusses the importance of financial analysis and the role of the accounting department in providing insights into a company's financial performance. It outlines the various techniques used to analyze financial data and the various factors that can affect a company's financial performance.

7. The seventh part of the document discusses the importance of financial control and the role of the accounting department in ensuring that all financial transactions are properly recorded and controlled. It outlines the various internal controls that a company may implement and the various methods used to monitor and evaluate these controls.

8. The eighth part of the document discusses the importance of financial compliance and the role of the accounting department in ensuring that all financial reporting is in compliance with applicable laws and regulations. It outlines the various requirements that must be followed in preparing financial statements and the various consequences of non-compliance.

9. The ninth part of the document discusses the importance of financial communication and the role of the accounting department in providing clear and concise financial information to management and other stakeholders. It outlines the various methods used to communicate financial information and the various factors that can affect the effectiveness of this communication.

10. The tenth part of the document discusses the importance of financial innovation and the role of the accounting department in adopting new technologies and methods to improve financial reporting and analysis. It outlines the various challenges that must be overcome in implementing financial innovation and the various benefits that can be realized from these efforts.

RECEIVED 2 : MAR 1972

MANUAL (NOTICE NO.)

4/1/72

4310

GEORGE 3 AND 4 OPERATING (4)

OXFORD UNIVERSITY COMPUTING LABORATORY
Copy 1 *Library*
COMPUTING SERVICE

File one copy of this notice with each of the manuals indicated.

LISTINCINDEX SYSTEM MACRO

Function

- 1 Lists :SYSTEM.INCINDEX on the line printer in an easily understood format
- 2 Lists any wronged tapes in :SYSTEM.INCINDEX

Format

- 1 LISTINCINDEX
- 2 LISTINCINDEX W

Execution

A job is run under the user :OPERATORS using :MACROS.LXJDF as a job description file.

If format 1 is used, the job loads and enters :DUMPER.PROGRAM G3DR (the program used by the macro LISTDUMP). The program lists a table showing the bit position, octal value and meaning for both the status (state of the increment) and magstate (state of the magnetic tape) words. It also produces a listing of :SYSTEM.INCINDEX giving, for each increment, the status, date, time, size (in 512 word blocks) and number of files followed by pairs of words for each tape that the increment is on, the first word of the pair being the tape serial number and the second the magstate word for that tape.

If format 2 is used, the job loads and enters :LIB.PROGRAM XKHS. This facility produces a listing of any wronged tapes in :SYSTEM.INCINDEX.

Files accessed

<i>File</i>	<i>Modes of access</i>
:MACROS.LXJDF	EXECUTE
:SYSTEM.INCINDEX	READ
:LIB.PROGRAM XKHS	EXECUTE
:DUMPER.PROGRAM G3DR	EXECUTE

Error messages

ERROR IN LISTINCINDEX: MACRO ABANDONED

PROPERTY NAMES PARAMETER

If more than one property name is to be specified, the names must be joined by ampersand (&). For example:

LISTFILE AF12,*LP,PAGE,PR PP120&FORM12&SLOW

THE PROPERTY SYSTEM (see page 29 of the manual)

If GEORGE tries to obtain a device on which to list a file and in so doing finds that there is no suitable device currently available owing to the absence or presence of some temporary exclusive properties, it will generate one of two system requests as follows:

- 1 PLEASE ATTRIBUTE *property name(s)₁* TO UNIT
peripheral description [WITH PROPERTY *name(s)₂*]
- 2 PLEASE FREE UNIT *peripheral description* OF TEMP
EXCL PROPERTIES

The first should be taken to mean that GEORGE requires the specified unit to have all of the properties listed in *property name(s)₁* attributed to it, and all other temporary exclusive property attributions should be cancelled. It is possible that some of the listed properties are already attributed to the unit in which case only the remainder need be attributed this time. The information in parentheses is a list of PERMANENT and CONSOLE properties that are attributed to the unit. If there are not such attributions, this part of the message is omitted.

The second message is self-explanatory and is output when no new attributions are required but extant TEMPORARY EXCLUSIVE attributions are preventing the file being listed on the specified unit.

CORRECTIONS TO USER NOTICE 1: LISTDUMP SYSTEM MACRO

Page 4, line 22: the sentence "It is not ... Execution)" is incorrect and should read as follows:

"It is only allowed in TTTP mode."

Page 6, line 24: the sentence "The parameters ... not specified." is incorrect and should read as follows:

"The parameters are analysed and :SYSTEM.INCINDEX is read and listed provided that the TAPE *magnetic tape description* is not specified."

MANUAL (NOTICE NO.)

8/3/72

4310

GEORGE 3 AND 4 OPERATING (3RD ED.) (5)

OXFORD UNIVERSITY COMPUTING LABORATORY

Copy 1

Library
COMPUTING SERVICE

File one copy of this notice with each of the manuals indicated.

MESSAGES FROM AN XKOF RUN

During a restore or compilation of GEORGE 3 (which is in fact a run of the GIN compiler), many operator messages are theoretically possible. The few which are likely to occur are given below. In general, any other message indicates an unusual situation such as corrupted tapes or a hardware failure. In such cases, the run should be begun again, from the beginning. It is always advisable to double check the cards in the restore pack in such cases.

Message	Meaning	Action
DISPLAY RP	The user Program File has been successfully restored.	None. This message always appears in a normal run.
DISPLAY RW	The user Workfile has been successfully restored.	None. This message always appears in a normal run.
DELETED OK	Successful end of run.	Go on to next job.
DELETED CC	#COR card has improper format.	Correct card and re-run job.
HALTED CL	GIN cannot get the core it requires.	Make more core available, and GO.
DELETED FI	The last directive listed is in error and compilation cannot continue.	Correct card and re-run job.
DISPLAY LP UNIT X FIX	GIN has disengaged the printer because 'paper low' indicator has been set.	Deal with the paper low situation and re-allocate printer. Program will continue automatically.

Message	Meaning	Action
HALTED PC	Disc file not available.	Either make required cartridge available and GO, or check card for mis-punching of filename or cartridge number, correct the card and re-run the job.
HALTED PL	Not enough drum to contain GEORGE file.	Clear drum and GO, or abandon job and re-load Executive clearing drum.
HALTED PO	Not enough drum or disc to contain GEORGE file.	Re-run with a larger value on the #PRO card.
HALTED PT	Disc or drum: There is a temporary hitch connected with the number of files open, or other programs having system files open, or integrity code failure.	Can happen either just after loading Executive or the disc cartridge on which the file is, or when multiprogramming. GO after a minute or two, or after other programs using same backing store have finished.
HALTED WI	Workfile invalid. The device type given in the #WOR directive is not acceptable.	Correct card and re-run job.
HALTED WT	Either too many files are open, or system files are open on the work-file device.	As for HALTED PT.
DELETED WU	Workfile used up.	Increase the size of workfile given in the #WOR directive or, if workfile is on disc, check that a sufficiently large scratch area is available. If it is not, load a cartridge with a larger scratch area. Re-run job.
DISPLAY WX DELETED AB	Happens if #COR card missing.	Supply #COR card and re-run job.

MEANING OF MESSAGES WHEN A DIRECT ACCESS DEVICE FAILS - MARK 6.3 ONWARDS

The messages output are:

UNIT *nn* FAIL (nn = Geographical number of device)
 TRACK **nnnn* BLOCK**nnnnnnnnnn* (E.D.S., EDS30 and F.D.S.)
 WORD **nnnnnnnnnn* (Slow drum only)
 BLOCK **nnnnnnnnnn* (Highspeed drum only)

These are followed by a print of the 5-word control area involved in the failing PERI order. (See below.)

The five word control area print may be followed by the message:

PLEASE RE-ALLOCATE

This is output if a read transfer from a filestore residence fails or if position fail is reported on an F.D.S. (see B4 of word 1 below). If the transfer was from an E.D.S. or EDS 30 (device types 6 and 26 respectively), GEORGE will disengage the device and the device need only be re-engaged for GEORGE to attempt the transfer again. For F.D.S. (2A and 2B) and slow drum the unit will have to be put off-line and then on-line manually by the operator. GEORGE will then attempt the transfer again. If a read failure occurs on a highspeed drum, the 5-word control area will be output and GEORGE TRANFAIL will be obeyed.

If a device becomes inoperable, the message:

UNIT *nn* FIX

is output. The device should be engaged and GEORGE will attempt another transfer. In this case no other information is output on the operators console.

CONTROL AREA INVOLVED IN THE FAILING PERI ORDER

If UNIT *nn* FAIL is output, the corresponding control area is as described below.

Word 0 Type/Mode

B0=1 Always. (Unit number in X of PERI X N(m))
 B1-B8 Device type
 B9-B23 Mode of transfer
 =0 Read from device
 =1 Write to device

Word 1 Reply word

E.D.S. and EDS 30

B0=1	Transfer in progress, other bits indeterminate	}	See Notes
B0=0	Transfer completed		
B1=1	Transfer terminated by end of cylinder		
B2=1	Track faulty		
B3=1	Uncorrected error		
B5=1	Inoperable		
B6-B23	Indeterminate, but see notes		

F.D.S. (2A AND (2B)

B0=1	Transfer in progress: other bits indeterminate	}	See Notes
B0=0	Transfer complete		
B3=0 } B1=1 }	Transfer terminated by end of cylinder		
B3=0 } B2=1 }	Address lost during data transfer		
B1=B2=0 } B3=1 }	Uncorrected error		
B3=0 } B4=1 }	Position fail on positioning		
B5=1	Device inoperable or rejecting transfers		
B6-B23	Indeterminate but see notes		

Additional replies for the 2B F.D.S. (2806) are as follows:

B1=1 } B3=1 }	Standard interface parity fail. (See notes)
B2=1 } B3=1 }	Multi-access failure: the device or channel cannot be accessed because it is allocated to another channel or processor. (See notes)

SLOW DRUM

B0=1	Transfer in progress: other bits indeterminate. (See notes)
B0=0	Transfer complete
B3=1	Error (Repeat procedure exhausted)
B5=1	Inoperable
B6-B23	Indeterminate

HIGH SPEED DRUM (2851)

B0=1 Transfer in progress: other bits indeterminate
 B0=0 Transfer complete
 B3=1 } Uncorrected
 B1=B2=0 }
 B3=B1=1 } Standard interface parity fail. (See notes)
 B2=0 }
 B3=B2=1 } Multiaccess failure: the device or channel
 B1=0 } cannot be accessed because it is allocated to
 another channel or processor. (See notes)
 B5=1 Device inoperable or rejecting transfers
 B6-B23 Indeterminate

NOTES:

The following replies will all result in GEOERR TRANFAIL after the control area has been printed:

B0=1 Unterminated transfer
 B1=1 End of cylinder (but not always, see below)
 B2=1;B3=1 Multiaccess failure (2B F.D.S. and HSD)
 B1=1;B3=1 Standard interface parity fail

GEOERR TRANFAIL will also be obeyed if a read transfer to a high speed drum fails.

B6-B23 are meaningless unless B1=1 in which case these bits contain the address of the word following the last word transferred.

The end of cylinder error implies that the hardware has transferred an incorrect number of words. If it is possible for GEORGE to recover from this situation the device is disengaged and then a request to re-allocate the device is output (as for normal read failures).

Word 2

Amount of data to be transferred in words

Word 3

Address of the first word of the buffer area

Word 4

(1988) 11000 00000 0000

Device address. (The message above the 5-word control area is this word converted).

E.D.S.

B2-B5 Most significant 4 bits of the head position
B8-B11 Least significant 4 bits of the head position
B14-B17 Head address
B21-B23 Sector address

F.D.S. 2A (2804 AND 2805) AND EDS 30

B0-B11 Head position
B12-B17 Surface
B18-B23 Sector

F.D.S. 2B (2806)

B0-B8 Head position
B9-B14 Surface
B15-B23 Sector

SLOW DRUM

B0-B23 Starting word of the drum area involved in the transfer.

HIGHSPEED DRUM (2851)

B0-B23 Starting block of the drum area involved in the transfer

CORRECTION TO MANUAL

Page 16: Restoring and loading GEORGE using the allocator/loader

The format of the DA and ED directive with storage unit serial number is not available in Mark 6. Only a file name may be specified with these directives in Mark 6.

RECEIVED 20 APR 1972

MANUAL (NOTICE NO.)

26/4/72

4310

GEORGE 3 AND 4 OPERATING (6)

OXFORD UNIVERSITY COMPUTING LABORATORY

Library

COMPUTING SERVICE

File one copy of this
notice with each of the
manuals indicated.

ADDITIONS TO USER NOTICE 3 - THE TAPE-TO-TAPE PROCESSOR

Page 9

Please alter the second sentence of item (d) to read:

"Any tapes for which this "depth" exceeds the value of the DEPTH parameter are marked as obsolete in INCINDEX."

Page 10

Please add the following notes after (j):

(k)

TTTP expands its core size dynamically, so it cannot be run as it stands with REALTIME ON. If REALTIME is ON and the program has to expand, it terminates.

If it is desired that REALTIME should be used with TTTP, the CORE parameter should be used to set the core size for TTTP to the largest size it is thought will be required. (The size can be deduced from looking at a typical TTTP monitoring file for the installation.)

(l)

The specification of TTTP includes the statement that it returns tapes which have become obsolete, automatically. This will only be done if TTTP finishes with the message TTTP - PROCESSING COMPLETED (message (f)(ii) below).

In all other cases, it is left to the discretion of the initiator of TTTP to decide later whether tapes made obsolete by TTTP should be released. If it is decided that they should be, this may be done by an OBEY COMFILE command in a job run under :DUMPER, or by a DUMP RETURN or DUMP REUSE command at the operator's console.

FORM 1/230/45(3.69)

- (m) The criteria which TTTP uses in order to determine whether to allow more increments to be written to a previously used output tape (if processing ONTO OLD) are fairly strict. First, any intervening increments between the increment contained on the old tapes and that specified by FROM must be obsolete or obsolescent (that is, containing no useful files). For example, if 38 is the last increment on the old tapes and FROM 45 ONTO OLD is specified then increments 39 to 44 must be obsolete (bit 4 of the status word set) or obsolescent (bit 18 of the status word set); otherwise TTTP will refuse to use the tapes.

Second, the tape must satisfy all of the following conditions:

(i)	It must be a GEO3PROCTAPE (BIT 19 SET)	All refer to the MAGSTATE word for the tape
(ii)	It must be of depth zero (BITS 22 AND 23 UNSET)	
(iii)	It must not be failing (BIT 0 UNSET)	
(iv)	It must not be wronged (BIT 4 UNSET)	
(v)	It must not be full (BIT 18 UNSET)	
(vi)	It must have the last increment on tape bit set (BIT 1 SET).	

So, it may well be found that the message (c)(vii) concerning not finding COPIES resumable tapes could occur quite often.

Page 11 Please add the following messages:

- (b)(vi) READ FAIL ON TAPE *tsn*
(c)(xii) WRITE FAIL ON TAPE *tsn*
(xiii) END OF TAPE *tsn* DETECTED
(d)(iii) TAPE ERROR (SEE ABOVE)
(following a transcription error)
(iv) TTTP FAILED TO POSITION TAPE *tsn*
(e)(xi) FROM IS *mmmm*; TO IS *nnnn*
(before message (e)(i))

Please replace message (e)(iv) with:

- (e)(iv) NO MORE CORE AVAILABLE - REALTIME ON?
(followed by message (e)(v))

Page 13

Please delete error numbers 908, 909 and 911 as these are replaced by the messages (b)(vi), (c)(xii) and (c)(xiii) respectively.

ALLOCATE COMMAND

With respect to note 4 in the specification, it should be noted that if a dump is taken when a file is not on a special residence and a general restore is subsequently done from that dump the file will not be put on a special residence, even if the special residence to which the file was originally ALLOCATED is now available. It will be necessary to ALLOCATE the file again.

WHATPER COMMAND

The following changes have been mended into WHATPER in Mark 6.4. This section supersedes that on *Storage unit information* in User Notice 3 to the *Operating Systems GEORGE 3 and 4* manual and User Notice 2 to the *GEORGE 3 and 4 Operating* manual.

Storage unit information

Storage unit information is output for UDAS devices that are engaged or being used by the system (for example, holding GEORGE file, SWAP file, etc.). It consists of a message indicating the state of the device:

UNIT *n* BEING USED BY SYSTEM

or

UNIT *n* ENGAGED (HOLD SET) [the (HOLD SET) sub-message being output if applicable] followed by a list of the serial numbers of storage units (SUSN) on the specified hardware unit, and, if applicable, a list of all files open on a particular storage unit (with their File Generation Numbers):

SUSN number	
FILES OPEN ON THIS UNIT ARE:	
FILENAME	F.G.N.
File 1	Gen 1
etc.	etc.

ALTERATION TO PAGE 27: FAILURES DURING OUTPUT

In Mark 6.4, pressing the HOLD button on a line printer during a LISTFILE will no longer cause an automatic error restart. Instead, the message

PLEASE ENGAGE UNIT *peripheral description*

is output. The operator then has three options:

- 1 He can engage the printer and the listing continues.
- 2 He can issue a CANTDO command. This causes the listing to be abandoned on the current output peripheral and restarted, when possible, on another peripheral of the required type.
- 3 He can issue a TERMINATE command. The listing is then abandoned.

ALTERATION TO PAGE 28: OUTPUT TO SPECIAL STATIONERY

In Mark 6.4 there is no longer a page throw after listing a SETUP file and before listing the main file.

© International Computers Limited, Reading 1972

Library

RECEIVED 12 OCT 1972

MANUAL (NOTICE NO.)

29/9/72

4310
4334
4345

GEORGE 3 AND 4 OPERATING (7)
GEORGE 3 AND 4 OPERATION MANAGEMENT (2)
OPERATING SYSTEMS GEORGE 3 AND 4 (2)

File one copy of this
notice with each of the
manuals indicated.

CONTENTS

The information in this User Notice refers to GEORGE 3 Mark 7.

INDEX TO NEW FACILITIES IN GEORGE 3 MARK 7	Page 1
SUMMARY OF NEW FACILITIES IN GEORGE 3 MARK 7	6

INDEX TO NEW FACILITIES IN GEORGE 3 MARK 7

The index indicates:

- 1 what new facilities are available
- 2 extensions or changes in existing facilities.

The titles of manuals given in the documentation references are abbreviated as follows:

G = Operating Systems GEORGE 3 and 4, Fifth edition
(TP 4345)

M = GEORGE 3 and 4 Operation Management, Fourth edition
(TP 4334)

O = GEORGE 3 and 4 Operating, Third edition (TP 4310), as amended by Amendment List 1, which will be published soon.

Chapters are referenced by numbers suffixed to the letters. Appendix is abbreviated App.; User Notices are referred to as G, M or O User Notice number, for example G User Notice 1. For example

CARTRIDGE command M2, G12, O3, O8

means that descriptions of this command can be found in:

GEORGE 3 and 4 Operation Management, Chapter 2.
Operating Systems GEORGE 3 and 4, Chapter 12.
GEORGE 3 and 4 Operating, Chapters 3 and 8.

Additional information describing 7020 and 7900 communications systems will be found in the manual *Data Communications and Interrogation*, Second edition (TP 4201).

This index does not reference every mention of a facility, but merely those thought to be useful for explaining Mark 7 facilities.

Accounting System	M10
ALLCHAR file type qualifier	G10
ALLOCATE PUBLIC Command	M13, G12
Allocator/Loader	M3, O2, O App1
ALWAYS DUMP GIN macro	M4, M App1
ARCHIVE command	M4, M13, G12
ASSIGN command	G3, G12
AUTMOP GIN macro	M App1
Associations on local peripherals	M11
ASSOCIATE command	M11, G12, O8
ATTACH command	G12, O8
Auto paper tape readers on MOP terminals	G7
AUTO qualifier to ASSIGN command	G12, G14
Autonomous exofile transfers	G User Notice 1
Autowrite output to offline *LP files	G12, G14, G App1
Backing store unjammer	M4
BACKTHRESH installation parameter	M4, M7
Basic files	G3
BILA GIN macro	M10, M App1
Break-in	G7
BUDGET command	G User Notice 1
Budget extracodes for System	
Journal processing	M User Notice 5
CANCEL ARCHIVE command	M13, G3, G12
CANTDO command	G12, O8
CARTRIDGE command	M2, G12, O3, O8
CATEGORISE GIN macro	M User Notice 5
Communication between operators' consoles (SPEAK command)	G12, O3, O8
Conceptual multiplexers on 7900 systems	M11, G5
CONCEPTUAL command	G12, O8
CONNECT command	G12
COPY command	G12
Core allocation system	G User Notice 2
CREATE command	G12
CRUNCHTIME GIN macro	M App1
DEAD command	G12, O3, O8
DEFAULTBS installation parameter	M7, G3
DETACH command	G12
Delay LOADING of object programs	G2
DIAGPROG system macro	M12, M App7
Direct Response Mode - basic peripherals	M11, G14

GEORGE 3 AND 4 OPERATING (7)
 GEORGE 3 AND 4 OPERATION MANAGEMENT (2)
 OPERATING SYSTEMS GEORGE 3 AND 4 (2)

Direct response Mode for exofiles	M11, G14
Disconnecting inoperable MOP lines	G7
DISCONNECT command	G12
Editor, H instruction	G6
Editor, O instruction	G6
Editor using workfiles in M and U instructions	G6
Editor, PS/--/ instruction	G6
Editor, Q instruction	G6
Editor, TS/--/ instruction	G6
EMPTY qualifier when ASSIGNing *DA and *DR output files	G12
ERASE command	G12
ERASE traps	G3
Error messages during General Restore	M4
Exchanging filestore cartridges	M2, G12, 03, 08
EXPRESS command	M8, M13, G12, 05, 08
File type qualifier for INPUT command	G3
FIND command macro	G12
FINISH command	G12
FS Allocator/Loader directives	M3, 02
FSRELOAD GIN macro	M User Notice 1
FULL monitoring file category	G10
FULLBUT monitoring file category	G10
General Destructive Reading of files	G3
General Restore	M4
GEORGE Mean Time	M User Notice 1
GEORGE Postmortem - second error	M User Notice 1
GEORGE system error - 7023 messages	M11, 06
GERONIMO subject program	O App2
GET command	M11, G12
GETONLINE command	M11, G12
GIN restores, *DA type GEORGE file on drum	M3, M App1
GRAPHIC file type qualifier	G10
HARDWARE command	M13, G12
High level scheduler	M8
High priority programs	G2, G12
IF ELSE command	G2, G12
Indexed serial files	G User Notice 1
Initial file traps	G3
Inoperable MOP lines	G7
INOPTIME GIN macro	G7, M App1

INPUT command	G12
Installation parameters	M7
Inter machine communication (IMC)	G5, G14
Inter program communication (IPC)	M11, G5, G14
INSTPARA command	M13, G12
IPCLIM GIN macro	M11, M App1
IPEDIT command macro	M13, G12
Journal Accounting Program (JAP)	M10
Limbo state for MOP jobs	G7
Limiting on-line backing store	G3
LISTFILE command	G12
LISTFILE command, SETUP option	G12
LISTDIR command	G3, G12
LOAD command	G2, G12
Log analysis	M10, G9
Logging messages	G9
:MANAGER, GROUP traps	G3
Manager-ARCHIVE	M10, G12
:MASTER.DICTIONARY	G User Notice 1
MENDOFF, MENDON GIN macros	M3, M App1
MOP, minor enhancements	G7
MOP command	G12, O8
MOP on operator's console	M App8
MT Librarian properties	M11, O3
MT traps	G User Notice 1
NEW command	M11, G12, O3, O8
NEWJOBDATA GIN macro	M App1
NEWMESS GIN macro	M User Notice 5
Non engageable exotic peripherals	M11, G5
NORMAL file type qualifier	G10
NOTOWNER privilege	M4, M10, G10
ONLINE command	G12
Operator's peripheral numbers up to 511	G14
OPTRACE installation parameter	M7
PAGES command	G12, O8
Paper tape handling	G3, G14
PARAM parameter with JOB, LOGIN, RUNJOB	G12
PERFWAIT installation parameter	M7, M10
PMPRINT system macro	M User Notice 3
PRINT command	G2, G12
PROPERTIES for Inter-processor communication	M11, G5
Public files	G3

GEORGE 3 AND 4 OPERATING (7)
 GEORGE 3 AND 4 OPERATION MANAGEMENT (2)
 OPERATING SYSTEMS GEORGE 3 AND 4 (2)

REALTIME...PRIORITY command	G2, G12
REDON command	G12, 08
Remote job entry from 1901A machines	M11, G5
REPORTTIME installation parameter	M7
RESET command	G12
REVGEN qualifier for exofile descriptions	G10
Revised generation number for ONLINE output *DA files	G12
RIGHT command	G12, 03, 08
SAVE command	G2, G12
SCRATCH qualifier for disc filenames	G10
Serial file types	G3
SETPARAM CORE command	G2, G12
SJMESS command	M User Notice 5
SJMESS extracodes	M User Notice 5
SJSIZE GIN macro	M User Notice 5
SPEAK command	G12, 03, 08
TAPE GIN macro	M11, M App1
STOPLIST command	G12, 03, 08
System Journal	M9
:SYSTEM.PERFORMANCE files	M10
:SYSTEM.OUTPUT	G User Notice 1
System requests without 'PLEASE'	01
:SYSTEM.SERIAL	G User Notice 1
TERMINATE command	G12, 03, 08
TRACE command	G12
TRANSERR GIN macro	M App1
TRANSPORTS command	M11, G12, 03, 08
Trap system	G3
TRAPSTOP entrant description qualifier	G3
UDAS mode PERIs for offline *DA files	G14
Unloading magnetic tapes	G User Notice 1
UNLOAD command	M11, G12, 03, 08
URGENCY command	M8, M11, G12
WHATJOBS command	G12, 05, 08
WHATPER command	G12, 03, 08
WHATPROPS command	G12, 08
WHATSTATE command	G12, 05, 08
Within-machine communication (WMC)	G5, G14
Write permit rings on magnetic tapes	G User Notice 1
WRONG command	G12, 03, 08

SUMMARY OF NEW FACILITIES IN GEORGE 3 MARK 7

These notes introduce the main new facilities of GEORGE 3 Mark 7. They are not specifications in any sense and the appropriate manual should always be consulted before using the facility. The index included in this notice will help users to find the description of an item in the relevant manuals.

New Executive interface

GEORGE 3 Mark 7 works to a new interface with Executive. Five main advantages are derived from the new interface:

- 1 GEORGE can now deal with more than 64 peripherals (including the operator's console).
- 2 Extracodes are now implemented more efficiently by Executive.
- 3 GEORGE can obtain information about peripherals more efficiently.
- 4 The information needed for timesharing object programs is no longer duplicated in GEORGE and Executive.
- 5 Object programs are handled more efficiently.

New and enhanced communication facilities

COMMUNICATION BETWEEN NOMINATED CONSOLES

The new operator command SPEAK (SK) allows communication between operators' consoles as follows:

- 1 Central operator to a remote operator
- 2 Central operator to all remote operators
- 3 A remote operator to the central operator
- 4 A remote operator to another remote operator

A specified text is sent to the destinations defined in routing parameters to the SPEAK command. The command should be very helpful when central and cluster console operators need to communicate with each other.

IMPROVEMENTS TO HANDLING OF 7900 SYSTEMS

Conceptual multiplexers

Conceptual multiplexers can now be defined on 7901/7903 message buffering systems. The facilities are identical to those implemented in Mark 6 for character

buffering devices.

Auto answering with 7903 systems

Auto answering via DATEL-200 modems is now available on 7903 message buffering systems.

USING 1901A MACHINES AS REMOTE JOB ENTRY TERMINALS

1901A systems may be used as Remote Job Entry terminals via character and message buffering multiplexers. In this role they are analogous to 7020 systems.

ENHANCEMENTS TO MOP FACILITIES

Automatic paper tape readers attached to termiprinters and some MOP terminals.

The paper tape output at MOP terminals may be re-input on MOP terminals and termiprinters that are fitted with automatic tape readers. The format of the tape is controlled by including the GIN macro AUTMOP in the restore pack. Strictly this macro is only necessary if 7900 systems are being used; however its use with character buffering devices will not have any undesirable effects.

Handling of inoperable MOP lines

If a MOP terminal becomes inoperable any job which is being run from it is DISCONNECTed and suspended rather than ABANDONed. A job in this state is said to be in *limbo*; it can be resumed by CONNECTing to it from another MOP terminal. When a job is in limbo an L is printed in the MOP? column in the output from WHATSTATE and WHATJOBS commands.

RESET command

The RESET WIDTH command is introduced to allow users to define the number of print positions available on a MOP terminal. This will be very useful where new devices which have more print positions than the standard 7071 MOP terminal are introduced, for example termiprinters.

Job handling

CORE ALLOCATION SYSTEM

An enhanced core allocation system is introduced. This will allow real time programs to be sited more efficiently and permit object program core utilisation to approach

the value defined by the installation parameter OBJECTQUOTA.

THE HIGH LEVEL SCHEDULER

The high level scheduler issued with Mark 7 is basically similar to that used under Mark 6 although there are several changes.

It is possible for an installation to reduce the overhead associated with the HLS by using it in a 'reduced facility' mode. Dictionary checking and the saving of tentatively started background jobs may be dispensed with. They are turned on and off with new parameters to the EXPRESS command. The state of job saving and Dictionary checking is remembered across HLS switches. This new facility allows users to trade off flexibility against overhead.

The EXPRESS command now overrides earliest start dates and times given via a JOBDATA command. EXPRESSED jobs are given URGENCY A without the user's budgets being checked when the job becomes fully started.

Any job which tries to LOAD a core image which is larger than (OBJECTQUOTA - size of HLS) is given a Computing Power Index of zero. Finally, jobs are not made fully started after a FINISH command has been issued.

HIGH PRIORITY PROGRAMS

Program members can be run at a priority which is higher than that of GEORGE. Normally these members will be controlling real time programs. A job will use a REALTIME ON, PRIORITY command to specify which members of a program are to have high priority status. Normally a program should not have more than one high priority member.

IF ELSE command

An IF *condition*, (*command*₁) ELSE (*command*₂) format is introduced. The commands may be nested to produce very powerful expressions.

INTER PROGRAM COMMUNICATION

Inter-Program Communication (IPC) permits one program to transmit data to another program using a type 28 I/O channel. The transmitting and receiving programs may be in the same machine which is called Within Machine Communication (WMC); or they may be in different machines connected by a 7210/1 IPB when they are using Inter Machine Communication (IMC). The connections are called IPC links and they are always simplex.

GEORGE 3 AND 4 OPERATING (7)
GEORGE 3 AND 4 OPERATION MANAGEMENT (2)
OPERATING SYSTEMS GEORGE 3 AND 4 (2)

There may be any number of IPC links open at any one time, and any number of links open through an IPB. The number of links may be controlled by the Installation manager with the IPCLIM GIN macro.

The IMC facility allows flexible multi-machine 1900 configurations to be created.

Two 1900 processors can be connected even if one is not running under GEORGE since the IPC facility provided by GEORGE is compatible with that provided by operator's Executive.

LISTFILE command

Several small changes are made in Mark 7. These are described below.

Listing serial files

Any serial file may be listed on any output peripheral. An attempt to LISTFILE certain system files or indexed serial files (all directories, :MASTER.DICTIONARY, :SYSTEM.OUTPUT and :SYSTEM.SERIAL) will generate a command error. When a file containing shift data is printed the data is converted to GRAPHIC form unless SPECIAL was supplied as a parameter to the LISTFILE command. In this case the data is printed in its full three shift internal format.

Long records

When a record which contains more characters than there are available print positions is listed on a line printer then the current line is truncated and the last four characters are a continuation mark: VV-V. The rest of the record is printed on subsequent lines, each of which is indented by four characters relative to the start of the first line. Records of more than 128 characters are not truncated when they are LISTFILED to a paper tape punch.

Peripheral interrupts

When the HOLD button, or the STOP button (other than the STOP button on a remote printer) is pressed or 'paper-low' is detected, GEORGE outputs a 'PLEASE ENGAGE....' message. If the device is engaged the listing continues from the current line. If a CANTDO reply is given the listing is abandoned on the current device, and GEORGE attempts to output it later. When the PAGE restart option is specified in the LISTFILE command 'PLEASE GIVE PAGES.....' message is output. The reply is used to calculate a suitable starting line when another attempt is made to list the file.

The TERMINATE command

When a LISTFILE is terminated during the listing of headings it will be repeated later. If a SETUP file is TERMINATED GEORGE assumes that the paper is correctly aligned and if it is an ONLINE request gives the device to the object program; if it is a LISTFILE request the main file is listed.

FINISH command

When FINISH NOW is typed any listing in progress is terminated and 'saved' for printing next time GEORGE is loaded. When FINISH 'time up' occurs no more new listings are initiated but current listings are completed. After all user jobs are complete any LISTFILES outstanding for system issued jobs, for example dumper, are output if suitable devices are available. Any remaining listings are 'saved' until the next time GEORGE is loaded.

LOAD command

In GEORGE 3 Mark 6 object programs occupy backing store and core store during the interval between a LOAD command and the first entry to the program. In Mark 7 the LOAD command will check the program's request slip. Core and backing store will not normally be reserved until the program is run by, for example, ENTER or RESUME. Any command which reads from or writes to the core image (for example PRINT, SAVE, ON, OFF, ALTER, CORE etc.) causes the LOADING process to be completed. This enhancement increases the efficiency of backing store and core use.

PRINT to file command

The PRINT command is enhanced to allow one or more regions of the current core image to be written to a named serial file. This may be created, overwritten or appended to by the user when the command is issued. The file can be a multifile. The file may be LISTFILED subsequently. The facility will be very useful when object program postmortems are to be preserved for later analysis. Large programs can be printed without filling up the job's monitoring file.

SAVE command

The SAVE command is changed to allow core images to be saved in a direct access file. Programs are loaded more efficiently from direct access files than from serial files. An optional second parameter to the command is introduced which defines the type of file to be used; it may be *DA, *ED, *FD or *CP. If *DA, *ED

or *FD is specified or the parameter is omitted a direct access file is used. If *CP is specified a serial file is used.

SETPARAM CORE command

A SETPARAM CORE command is introduced. This allows user jobs to determine the size of their current core image in words.

THE SYSTEM JOURNAL

The System Journal is a series of basic serial file-store files in which GEORGE stores information about what happens while it is running. Data is sent to the Journal in messages which can also go to several other places including a job's monitoring file, MOP terminals or the operator's console. The installation manager has some control over the destination of messages by using the GIN macro CATEGORISE. He can also vary their content with the GIN macro NEWMESS.

The messages contain various types of information (such as machine usage, accounting and engineers' data), in a form which is readily processable by subject programs. Information is stored in the Journal whenever a program alters its core size or a job changes its urgency or releases a peripheral. The data includes jobs' milltime, programs' clocked time, counts of transfers, fails and repeats for all attached peripherals and details of lines, pages and properties for each job's LISTFILES. It can be analysed by users' subject programs for a variety of purposes such as accounting, analysis of machine usage, and engineering reports.

The flexibility of the message system is further increased by the provision of a new command SJMESS (and two equivalent extracodes) which will write messages specified by privileged users into the Journal. The command can be issued from the operator's console, MOP terminals or background jobs. The extracodes will send messages from users' object programs.

The main use of the System Journal by ICL in Mark 7 is to provide flexible, efficient accounting. This is accomplished by the Journal Accounting Program (JAP), a processing program which is run at system-determined intervals to analyse Journal files, produce accounting data, update budgets in the Dictionary and print invoices. System Journal accounting is initiated by including the GIN macro BILA SJ in GEORGE's restore pack. This switches off built-in and built-out log analysis and permits all charges to be calculated by the Journal Accounting Program from its log analysis. In its issued form the JAP uses the same algorithms as the existing built-out log analysis program #XK7F, but provision is made for user alteration of charging

co-efficients.

The Journal Accounting Program is run as an ordinary job under :JOURNAL. Every time the JAP updates the Dictionary it inserts an 'update record' in the Dictionary. This contains all the information necessary to restart the program at the right place in the System Journal, so the JAP can determine how much of the Journal it has already processed. This prevents jobs being accounted for twice and provides protection against breakdowns or the JAP being abandoned.

GEORGE also has special mechanisms which protect the Journal files against system breaks. Mark 7 allows parallel running of existing built-in or built-out log analysis systems and System Journal accounting. This simplifies the validation of the new accounting methods when they are adopted by each installation.

Enhancements to the filestore

ARCHIVE COMMAND

This new command can be used to indicate that a file may be thrown offline by the Backing Store Unjammer. The command is 'permanent' so a CANCEL ARCHIVE command is provided to de-ARCHIVE the file. A NOTOWNER privilege is introduced. Users possessing this privilege may ARCHIVE files belonging to other users. For instance, the installation manager can ARCHIVE files that he considers should be removed immediately a Backing Store Jam occurs. Files which are defined in an ARCHIVE command have indicator bits set in their directory. This allows user accounting systems to charge separately for these files.

AUTOWRITE OUTPUT TO OFFLINE *LP FILES

The filestore line printer file handling routines are enhanced to accept object program line printer output in Autowrite format. The offline files produced in this way are then LISTFILED producing the same effect as using Autowrite mode and an ONLINE buffered line printer, although the LISTFILE routines themselves do not use Autowrite transfers. The ASSIGN command for the line printer file must have an AUTO qualifier.

BASIC FILES

In GEORGE 3 Mark 6 there can be as many basic serial file types in the filestore as there are basic peripheral types. The type of the file corresponds to the peripheral from which it was INPUT or the offline basic peripheral channel on which it was output. Hence the filestore can contain card reader

files, card punch files, tape reader files, tape punch files and line printer files. The handling of basic files is simplified and rationalised in Mark 7.

Information handled via basic peripherals is called basic data; it consists of a sequence of records each of which is a string of characters. There are three types of basic data, GRAPHIC, NORMAL and ALLCHAR. GRAPHIC data is composed of the standard ICL 1900 series 64 character set, NORMAL and ALLCHAR data is composed of the 1900 series shift character set which includes upper and lower case letters and various control characters.

ALLCHAR data includes the entire 128 character set whereas the NORMAL set excludes runout and delete characters. Note that these data types describe the external character set and not the way any given type of data is represented on a particular medium or internally by GEORGE.

There are only three types of basic files, GRAPHIC, NORMAL and ALLCHAR which correspond to the three types of data already described. In contrast to Mark 6 a file's type is determined by the type of data it contains and not by the device or channel originally associated with it. Much data is GRAPHIC and users are recommended to use this type of file wherever practicable because it employs a less complex internal representation than the others, and can be manipulated more readily.

The file type qualifier

A file's type may be explicitly declared with a qualifier to the filename specified in an INPUT command. This qualifier can be GRAPHIC, NORMAL or ALLCHAR, for example:

```
IN :MG,ABC (GRAPHIC)
```

If the qualifier is omitted the file will be of GRAPHIC type by default. A similar qualifier may also be used when offline basic output files are ASSIGNED for writing.

CHANGES TO FILE TRAPS

An ERASE trap is introduced. The ERASE and RENAME commands check this trap; the ERASE command no longer checks for ownership except when attempting to ERASE a directory. If the trap is not open a command error is generated. A user can now allow other users to ERASE his terminal files; this facility is entirely separate from that of allowing other users to write to them.

When a file other than a directory is created all traps are initially left open to the owner of the file. In contrast, under Mark 6 the READ, APPEND and EXECUTE traps are initially left open and the WRITE trap is closed. Treatment of directories, however, is unchanged from Mark 6. A TRAPSTOP qualifier is introduced for relevant commands to allow users conveniently to reset the default traps.

The first change will increase the security of files by making it more difficult for users to accidentally ERASE their files, for example by typing ER *filename* instead of ED *filename* during a MOP job. The second change will slightly reduce the security of a newly-created file. However there is likely to be a reduction in the number of jobs which fail because the user has forgotten to open a file trap, whilst the TRAPSTOP qualifier allows users readily to define the initial level of security that they need.

GENERAL DESTRUCTIVE READING OF FILES

Files may now be accessed in General Destructive Read mode. This means that the records they contain are destroyed as they are read. A file may have one reader, multiple readers, single reader and/or multiple readers and communal writers using the file in communication mode. General Destructive Read can be used in combination with Direct Response Mode. A new qualifier to the ASSIGN command, GDR, is introduced.

GENERAL RESTORE

More informative messages are output if a magnetic tape fails or a tape format error is found during a general restore. Previously tape fails would force GEORGE to be reloaded and the Early Morning Start procedure to be repeated. If a format error occurred GEORGE would break GEOERR IDFORMAT. In Mark 7 messages are output giving the last file read, and stating the type of error. After an error the General Restore routines will release all the backing store already obtained and ask the operator to try again with a different increment and/or a different tape deck.

INDEXED SERIAL FILES

In Mark 7 certain commonly used serial files are indexed. The index is held in a file's directory entry and allows records to be accessed directly rather than serially, reducing the number of backing store transfers required and hence access times; the format of the file itself is not altered. The files may be processed serially or records can be accessed directly by calculating the record's position in the file from

GEORGE 3 AND 4 OPERATING (7)
GEORGE 3 AND 4 OPERATION MANAGEMENT (2)
OPERATING SYSTEMS GEORGE 3 AND 4 (2)

a key. In Mark 7 the following files are indexed: directories, :MASTER.DICTIONARY, :SYSTEM.OUTPUT and :SYSTEM.SERIAL. Although access to these files will be much faster users are not directly affected except that any attempt to INPUT, ASSIGN for writing, COPY to or EDIT an indexed file will result in a command error, and any programs which access directories using the File Reader may need modification.

LIMITING ONLINE BACKING STORE

The online backing store of a job is the total amount of space occupied by files used by the job at any given time. Because these files are associated with a job they cannot be discarded by the Backing Store Unjammer. By limiting the maximum online backing store used by each job the likelihood of severe backing store jams is reduced. A new installation parameter DEFAULTBS is introduced which defines in K words the maximum amount of backing store allowed to each job. If a job exceeds DEFAULTBS the operators are asked to increase the limit by changing DEFAULTBS. If they do this the job will continue or they may refuse, in which case a command error or program event occurs. Alternatively they can decide to ABANDON the job.

So-called public files are excluded from a job's online backing store count. Basic system files and directories are always public. Other commonly used files (for example compilers) may be declared as public with the ALLOCATE PUBLIC command which can be issued from the operator's console or by a job running under :MANAGER.

PAPER TAPE HANDLING

Paper tape input and output is simplified in Mark 7. The default file type is GRAPHIC, and, if a non-graphic tape is INPUT, a NORMAL or ALLCHAR entrant description qualifier must be included in the INPUT command.

UDAS MODE PERIS TO ACCESS FILESTORE FILES

UDAS facilities are provided for disc type filestore files. In Mark 6 UDAS modes are illegal when addressing filestore files. New modes are *1200 (Create and Open) and *1300 (Close and Delete), whilst meaningful action is implemented for modes *1100, *1400 and *1500.

UNJAMMING ALL RESIDENCES

When the Backing Store Unjammer is called in it will clear any jammed residences and examine all the unjammed residences. Files are removed from the latter if the residence is nearly jammed, that is, the percentage of used blocks is greater than a threshold value. This is determined by the setting of a new installation parameter BACKTHRESH, and is equal to $(BACKJAM - BACKTHRESH)$. Suppose that $BACKJAM = 80\%$ and $BACKTHRESH = 10\%$; then the threshold = 70% . The unjammer will throw files off any residences which are more than 70% used.

Special residences are no longer automatically unjammed. Jams on special residences are considered to be totally undesirable and they should be dealt with directly by the Installation Manager.

Peripheral Handling

ASSOCIATIONS ON LOCAL PERIPHERALS

The Property system is extended to allow Associations to be created for local peripherals. Previously this was not possible. By increasing the flexibility of the property system special applications such as the creation of dedicated local card reader and line printer combinations are possible. When a local card reader or paper tape reader is associated with a line printer or a card or tape punch, jobs initiated from the reader automatically send their LISTFILES to the Associated output device. Hence a 'Cafeteria' service for small jobs using, for example, ICL's FLAIR or BABS software, can be readily devised.

DIRECT RESPONSE MODE

In Mark 7 Direct Response Mode peripheral handling is made available for ONLINE exofiles and basic peripherals.

EXCHANGING FILESTORE CARTRIDGES

The CARTRIDGE (CG) command eliminates the need to break GEORGE when an EDS spindle holding a filestore or swap residence fails and it is necessary to transfer the cartridge on that spindle to another spindle. The operator types

CARTRIDGE Un_1, Un_2

here Un_1 and Un_2 are the operator's unit numbers of the failed spindle and the new spindle. The order of the parameters is immaterial. The two units are disengaged and the operator can transfer the cartridge from the failed spindle.

REVISED GENERATION NUMBER WHEN ONLINING EXOFILES FOR OUTPUT

A new qualifier to the exofile name, REVGEN, is available when a UDAS device is ONLINED for writing. The REVGEN qualifier allows the generation number of the output file to be revised, providing that the new number is <4096. The action of the ONLINE command and its equivalent open mode (#300) PERI instruction is now compatible.

UNLOADING MAGNETIC TAPES

After the system has finished using a tape the tape is rewound, the deck becomes inoperable and the deck address light goes out, indicating that the deck is not available to GEORGE. The behaviour of 1974 decks is not altered.

MT LIBRARIAN PROPERTIES

Magnetic tapes and decks can be described properties. These are:

- 1 Tape characteristic which is the recording method used to write a tape, for example seven track, nine track and/or phase encoded or NRZI.
- 2 MODE which describes the parity, gap and recording density used to record data on a 7 track tape.
- 3 Peripheral properties, encompassing the tape characteristic, speed properties and MODE.

The properties are specified in peripheral property parameters or qualifiers, or entrant property qualifiers, depending upon the context. This enhancement will allow the user and operators to have more control over the use of the available transports in installations with more than one type of tape transport. Enhancements are introduced to GEORGE requests for magnetic tapes, the UNLOAD, TRANSPORTS, GETONLINE, ONLINE, NEW and GET commands and unanticipated open mode PERI program instructions.

Operators' facilities

THE GERONIMO SYSTEM

The GERONIMO system allows some operator information (for example details of jobs currently running, files to be listed, etc.) to be output on one or more Mark II local alphanumeric visual display units (AVDU). The system comprises a command language macro, VIDEO, and a small realtime subject program which controls the AVDUs. The operator information is output in response to commands to GERONIMO input via the AVDU. Some

commands include an AUTO facility which causes the displayed information to be updated and refreshed periodically. Operators can also input and run small background jobs from the AVDU unit. A Master AVDU may be nominated in a configuration which has two or more AVDUs, (up to a maximum of six). The Master AVDU is provided with an extended repertoire of instructions to control the complete GERONIMO system.

A GEORGE 3 user may run GERONIMO if he has Mark II visual display equipment available, that is 7181/4 terminals connected via a 7180/2 cluster control unit.

MOP ON OPERATORS' CONSOLES

MOP facilities are provided on central and cluster operators' consoles. A small subject program, #XKLY, uses an online Command Issuer channel to set up and drive a MOP job and internally issued DISPLAYs and QUESTIONs to output data to, and read commands from, the operator's console. Break-in facilities are provided by a second system job which controls another subject program #XKLZ. This also uses a Command Issuer channel to CONNECT to the first job and to cause it to issue a Command Issuer break-in.

The facility will be especially useful to installations or remote clusters which do not have MOP terminals. For example, an operator can CONNECT a background job which is thought to be looping and PRINT an object program to the job's monitoring file rather than just ABANDONing it, leaving the user with no diagnostic information.

PMPRINT COMMAND MACRO

Post mortems are now dumped to magnetic tape in compressed line printer image format. The PMPRINT command macro, in conjunction with program #XK65, provides a simple method of listing these postmortems to a line printer or a line printer file. The command is issued from the operator's console.

c International Computers Limited, Reading 1972

RECEIVED 5 FEB 1973

Adv.

MANUAL (NOTICE NO.)

31/1/73

4310

GEORGE 3 AND 4 OPERATING (8)

File one copy of this
notice with each of the
manuals indicated.

AMENDMENT 1

This amendment is being issued in the form of a reprint of the manual incorporating the amended text; it constitutes the version of the manual which should be used with Mark 7 of GEORGE. This amendment supersedes User Notices (1) to (6), which may be discarded. User Notice (7), containing the Index and Summary of new facilities in GEORGE 3 Mark 7, should be retained.

Any user who is still using Mark 6 of GEORGE should retain the previous version of the manual and User Notices (1) to (6).

Blank pages

The following pages between command specifications in Chapter 8 have been left blank for production reasons:

70, 74, 122, 128, 144, 152

Command specifications

Users may be interested to know that the following commands have specifications specially written for this manual:

ANSWER	DISENGAGE	INPUT	RUNJOB	VIDEO
ATTACH	DOCUMENT	MODE	TAPERIGHT	WHATSTATE
DEAD	IDFLIST	PMPRINT	TAPEWRONG	

The specifications of all other commands are reproduced from the manual *Operating Systems GEORGE 3 and 4* fifth edition, TP 4345, with occasional minor changes. In these commands amendment lines in the margin reflect changes from the Mark 6 version of the latter manual.

© International Computers Limited, Reading 1973

1 of 1



Library

RECEIVED 19 APR 1973

PUBLICATION (NOTICE NO.)

11/4/73

4310
4345
4334

GEORGE 3 & 4 OPERATING (9)
OPERATING SYSTEMS GEORGE 3 & 4 (4)
GEORGE 3 & 4 OPERATION MANAGEMENT (12)

File one copy of this
notice with each of the
publications indicated.

CONTENTS

	Page
MOP ON MARK 2 VISUAL DISPLAY UNITS (7181)	1
THE VIDIMOP SYSTEM MACRO	4
MOP ON OPERATOR'S CONSOLE	9
NOTE ON USER NOTICE 8 TO GEORGE 3 & 4 OPERATING	12

MOP ON MARK 2 VISUAL DISPLAY UNITS (7181)

With GEORGE Mark 7, MOP facilities are made available on 7181/4 VDUs connected through a 7180/2 local VDU controller and on 7181/2 VDUs connected via a 7901 or 7903 communications processor.

This facility will shortly be made available using the following software:

VIDIMOP system macro
XKL4 program
XKL2 program

An announcement of the availability of this software will be made in a software notice at a later date.

Each VIDIMOP job will provide MOP facilities on up to 64 displays, but separate jobs must be run if both 7181/4 and 7181/2 displays are to be used simultaneously.

Because the MOP facilities are provided by a subject program, using anticipated command issuer (type 60) PERIs, rather than being built into GEORGE, there are certain minor differences from the standards outlined in Chapter 7 (*Using MOP*) of the manual *Operating Systems GEORGE 3 and 4* (TP 4345). These differences are outlined in the following paragraphs, which should be read as an addendum to that chapter.

The 7181 Visual Display Unit

The table below shows how the various device dependent features of MOP are implemented on the 7181 visual display unit.

1 of 12

<i>Signal description</i>	<i>Keyboard typing keys</i>	<i>Notes</i>
Start of message	None	
End of message	SEND	As for 7071 terminal
Ready		As for 7071 terminal
Message cancel	None	(These functions (are not requir- (ed as the hard- (ware editing (facilities of (the display per- (mit the user to (alter his input (at will before (sending, 'End (of message'.
Character cancel	None	
Break-in	INTERLOCK, CONTROL and 1	On keyboards with a shift- numerics block wired as an action key block the sequence INTERLOCK, action key 1 may be used as an alternative. The INTERLOCK key is the unlabelled key at the right hand end of the top row of keys, in the left hand block of keys.

Notes

- 1 The keyboard characters $\_$ and $\wedge$ correspond to the characters $\leftarrow$ and $\uparrow$ respectively in the 1900 series 64-character card code, and will be interpreted as such by GEORGE. These characters are displayed as -- and $\wedge$ respectively. Since there is no character cancel the $\_$ character will not be treated in any special fashion by GEORGE. The characters $|$ (vertical bar) and $\_$, if keyed in, will be discarded and ignored by GEORGE as they have no equivalents in the 1900 Series 64-character card code. Input may be in upper or lower case characters, but in the current mark all lower case alphabetic characters will be

GEORGE 3 & 4 OPERATING (9)
OPERATING SYSTEMS GEORGE 3 & 4 (4)
GEORGE 3 & 4 OPERATION MANAGEMENT (12)

interpreted by GEORGE as their upper case equivalents (that is, shift characters are discarded) and all output will be in upper case.

- 2 The maximum number of characters that may be input to GEORGE on a 7181 display unit in the current mark is 511 internal six-bit characters. If more than 511 characters are input a message

MESSAGE TOO LONG - PLEASE REVISE

is sent to the display, and the input must be reconstructed and sent again. (No more of the original message is overwritten than is absolutely necessary, so that this reconstruction is kept to the minimum.) It should be noted that a change of alphabetic shift occupies one internal six-bit character and `1, #, ^, _`, `|` (vertical bar) and characters occupy two internal six-bit characters, but the characters generated by the SEND key are excluded from the 511 count.

- 3 When the MOP service provided by VIDIMOP is started up the screen is cleared and the message:

MOP SERVICE READY AT *time*

is output for each operable display. If a previously inoperable display becomes operable the same occurs. When a command is issued which results in the MOP job returning to NO USER context the message:

CLOSED DOWN AT *time*

is sent to the display instead of the next ready signal.

After either of these events the user should break-in in order to get a new introductory message and invitation to type.

- 4 After the MOP SERVICE READY message, output and input occupy successive lines of the screen until the bottom line is reached. Further output or input results in the contents of the screen being moved up the screen, so that lines are lost from the top and room is made for further messages at the foot of the screen. This movement is known as *rack-up*. Lines lost in this fashion cannot be recalled. When a ready message is sent to the display at least two blank lines will be left for input. If the input message is so long that it is impossible to fit it into this space, but it is still less than 511 internal 6-bit characters (see note 2 above), it is permissible for the message to "wrap-round" to the top of the screen, but lines input at the top of the screen in this fashion will be lost as soon as rack-up occurs.

- 5 If the user wishes to suspend GEORGE output while the contents of the screen are examined, a "suspend output" command may be issued. This is similar to the break-in sequence except that CONTROL and 2, (or action key 2) are pressed instead of CONTROL and 1. In order to continue GEORGE output, either break-in or the "continue output" command (CONTROL and 3) must be issued.
- 6 If GEORGE sends 21 lines of output to the display without sending a ready message (under some conditions this may in fact be up to 23 lines) it suspends output as if the user had issued a "suspend output" command. The suspension is cleared in a similar fashion, by breaking-in, or issuing a "continue output" command.
- 7 There is no double break-in facility, whereby unwanted output is discarded.
- 8 If a display becomes inoperable the MOP will not, in the current mark, be put into a state of limbo, but will be abandoned immediately, and the message

JOB ABANDONED - CHANNEL RELEASE

will be sent to the job's monitoring file.

- 9 Since the MOP facilities on 7181 displays are provided by a subject program rather than by built in routines, no system breakdown message will be output in the event of a GEORGE failure. In the event of a subject program error, or the operators abandoning the VIDIMOP session, an attempt will be made to output the message

MOP SERVICE TEMPORARILY SUSPENDED

to each operable display.

THE VIDIMOP SYSTEM MACRO

Function

Provides MOP facilities on 7181/4 visual display units (connected via a 7180/2 local VDU control unit) and on 7181/2 visual display units (connected via a 7900 series communications processor).

Format

- 1 To run MOP via a local cluster control unit:

VIDIMOP *ID, JTtime, *CRfile description

GEORGE 3 & 4 OPERATING (9)
OPERATING SYSTEMS GEORGE 3 & 4 (4)
GEORGE 3 & 4 OPERATION MANAGEMENT (12)

2 To run MOP via a communications processor:

VIDIMOP *CC, JT*time*, CL*conceptual name*, *CR*file*
description

The JT, *CR and CL parameters are optional and may occur in any order.

The JT parameter specifies the jobtime for the job. This parameter is not relevant when an *CR parameter is present. *time* has the same format as the milltime parameter in a JOBTIME command.

The *CR parameter indicates an initialisation (or re-configuration) run (see *Execution*, below). *file description* defines a basic file which contains details of the displays to be addressed. If this parameter is absent a normal MOP run is initiated.

The CL parameter is only permissible in format 2. If it is absent a 7900 series communications processor will be ONLINED to the subject program; otherwise the conceptual communications processor named will be ONLINED. This parameter is not relevant when an *CR parameter is present.

Forbidden contexts

None, but not recommended in MOP & USER context.

Execution

Before the system can be put into everyday use the displays which are to be polled must be defined in an initialisation run, described in section 1 below. Thereafter normal runs adopt the procedure described in section 2, but to define a new configuration to be polled another initialisation run must be made.

1 The initialisation run

The macro runs a job IDVSETUP (format 1) or CCVSETUP (format 2) under :LIB. This job loads :LIB.PROGRAM XKLn(1), and assigns to it the file named in the *CR parameter. The program reads in parameter records until it reaches a terminator and is then SAVED into :LIB.PROGRAM XKLn(+1). The operator is then informed that the initialisation is complete.

The format of the parameter file differs between the local and remote programs:

For the local case (VIDIMOP*ID), the file must contain one record for each display to be used. The format of each record is #*xyyy* where *xyyy* is the octal representation of the hardware address of the display, for example #0103. This address

may be ascertained from the engineer: *xx* corresponds to the channel number of the cluster control unit, and is in the range 00 to 17; *yy* is the specific hardware address of the display, in the range 01 to 17.

The last record in the file must be ****.

For communication with the operator about displays the program assigns each display an identifier number (numbering the displays consecutively from 1) in the order specified in the parameter file. The correspondence between displays and identifier number is output to the job's monitoring file, and it is suggested that the listing of this file be retained for reference by the operator/engineer.

For the remote case (VIDIMOP *CC), each record in the file defines a separate identifier or group of identifiers and takes the form *n* or *n(m)*, where *n* is a decimal number in the range 1 to 4095, *m* is a decimal number in the range 1 to 63, and *n + m* < 4096. Identifiers must be declared in numerical order. For example,

```
1
3
2155(9)
```

defines identifiers 1, 3, 2155 to 2164

The last record in the file must be ****. Each record must start at column 1; leading and embedded spaces are not permitted.

Each parameter record is re-output to the job's monitoring file as it is read, together with, in the local case, the identifier number assigned to the display.

2 Normal runs

The macro runs a job IDVMOP (format 1) or CCVMOP (format 2) under :MANAGER. (The username :MANAGER may be changed by suitably editing the macro VIDIMOP, but the new user must have REALTIME budgets if the local (*ID) version of the macro is to be used). This job loads the version of :LIB.PROGRAM XKL*n* with the highest generation number, ONLINES the appropriate communications device or conceptual and enters the program.

If a display becomes inoperable, having previously been operable, or vice versa the job outputs the appropriate informatory message:

```
IDENTIFIER n UNAVAILABLE
or IDENTIFIER n AVAILABLE
or IDENTIFIER n OFF TEST (*CC version only)
```

GEORGE 3 & 4 OPERATING (9)
OPERATING SYSTEMS GEORGE 3 & 4 (4)
GEORGE 3 & 4 OPERATION MANAGEMENT (12)

If output to a display fails the job outputs the message:

IDENTIFIER n ERROR

and attempts the output to the display again.

If the communications device fails the job outputs the message:

*ID (or *CC) INOPERABLE: SESSION ABANDONED

and finishes.

If the operator wishes to terminate the MOP session he should CONNECT to the VIDIMOP job and issue the command "ON 2", then disconnect. The program will then output the message:

MOP SERVICE TEMPORARILY SUSPENDED

to each operable display, release the communications device, display

VIDIMOP SESSION FINISHED

to the operators console, and the job will end.

WARNING

The operator may not CONNECT to IDVMOP or CCVMOP from a VDU running under the system, as this will result in an irreparable deadlock in which neither the MOP job nor the control job can be ABANDONed, but he may CONNECT from any other MOP device.

If any other failure occurs the job will take a postmortem print of the program then attempt to output the message:

MOP SERVICE TEMPORARILY SUSPENDED

to all operable displays, and release the communications device.

When this has been done the operator is asked whether he wishes to continue the run. If the answer is YES, the job initiates another run, jobname :MANAGER.IDVMOP1 (or CCVMOP1), then ends. If the answer is NO the job ends without attempting to start a new run.

The monitoring file listing of the job should be examined to determine whether the fault can be corrected; if not the listing should be submitted through the normal GEORGE error reporting channels.

Library entrants accessed

:MACROS.VIDIMOP is used as a job description file.
:LIB.PROGRAM XKL4 is loaded to control 7181/4 VDUs
(VIDIMOP *ID)
:LIB.PROGRAM XKL2 is loaded to control 7181/2 VDUs
(VIDIMOP *CC)

Examples

VIDIMOP *ID,*CR:MANAGER.DISPLAYS

VIDIMOP *ID,JT300MINS

VIDIMOP *CC,CLDISPLAYS,JT3600

VIDIMOP *CC

Error messages

These are generated by DISPLAY commands within the job description, and are self-explanatory.

Notes

- 1 In installations which do not wish to run VIDIMOP on both a local cluster control unit and a message buffering multiplexer, it is possible to edit the VIDIMOP macro to preset the *ID/*CC parameter, so that the operator need only type VIDIMOP to run the job after the initialisation run has been completed.
- 2 If the remote (*CC) version is being used with a conceptual, identifiers may be ATTACHed to and DETACHed from the system at any time.
- 3 It is possible to have one local (*ID) macro and one remote (*CC) macro being obeyed by the system at the same time. If it proves necessary to run two or more local, or remote, controlling jobs simultaneously (for example, where there are two cluster control units, or where the routing of listfiles, as described below, is required for some displays but not for others) this is made possible by issuing a RUNJOB command to initiate the second, or subsequent VIDIMOP job, rather than using the VIDIMOP macro itself. The format of this command should be:

RUNJOB *jobname*, :*username*, VIDIMOP, PARAM (*ID, JT*time*)

or

RUNJOB *jobname*, :*username*, VIDIMOP, PARAM (*CC, KT*time*
CL*conceptual name*)

GEORGE 3 & 4 OPERATING (9)
OPERATING SYSTEMS GEORGE 3 & 4 (4)
GEORGE 3 & 4 OPERATION MANAGEMENT (12)

where *jobname* is arbitrary

username will normally be :MANAGER

the JT and CL parameters are optional

- 4 Response times on 7181/2 displays may be enhanced by making the controlling program REALTIME; this can be done by editing the VIDIMOP macro.
- 5 If the VIDIMOP macro (or RUNJOB) is issued from a 7020 console, acting as a cluster console, any listfiles initiated by VDU MOP jobs will go to output devices (if any) in the remote cluster, unless specifically directed elsewhere.

MOP ON OPERATOR'S CONSOLE

MOP facilities on the operators' console are provided in Mark 7 by a background job (hereafter referred to as the control job) which communicates with the operator by means of DISPLAY and QUESTION commands. The job runs a subject program which uses an anticipated command issuer channel to issue the operator's MOP commands to GEORGE.

Break-in facilities are provided by running a second background job. This job runs a program which CONNECTS to the control job, notifies the control job of the break-in request (the control job will then request commands in BREAKIN context), DISCONNECTS the control job and ENDJOBS.

Running a MOP job from the operator's console

The operator must initiate the control job, which uses the job description file OPSMOPJD, by issuing a RUNJOB command. The username given in this RUNJOB command need not be the same as that to be used for the MOP job.

The control job loads the subject program, opens a command issuer channel and DISPLAYs the output from this channel to the job's cluster console (that is, ordinarily to the operator's console, local or remote, from which the RUNJOB was issued). If any output message from the MOP channel is longer than may be accommodated in the text of one DISPLAY (that is, 40 characters) it is continued in further DISPLAYs (the message is broken at a word boundary, if possible), and for output containing commas text of the DISPLAY will be enclosed by quotes (").

Ready messages are sent using the QUESTION command, rather than DISPLAY, and the operator issues his MOP commands as the text of ANSWERS to the control job. Where the MOP command contains commas this text must be enclosed in brackets; the subject program will remove these brackets before issuing the command through the MOP channel to GEORGE.

For any % character (that is, parameter substitution) in the MOP command the text of the ANSWER must contain "%" as one of these %s is removed by the command processor in processing the parameters of the ANSWER command. Failure to double up on %s in this manner will result in the ANSWER command being errored:

ERROR IN VERB:PARAMETER SUBSTITUTION NOT ALLOWED
AT THIS LEVEL

Effecting a break-in

If the operator requires to break-in on his job he must initiate a further background job, using the job description file OPSBIJD. In this case the RUNJOB command must give the job three parameters: the jobname and username of the control job and that user's password. These parameters are provided so that the job can use a command issuer channel to CONNECT to the control job.

The username under which the break-in job is run need not be the same as that used for either the MOP job or the control job.

If the break-in job manages to force a break-in it will DISPLAY: BREAKIN. If the job finishes without any such DISPLAY the job has failed to CONNECT to the control job; the normal cause of this is some error in the parameters giving the identity of the control job. Under exceptional circumstances the break-in may fail even though BREAKIN has been DISPLAYed. If this does happen a further DISPLAY: BREAKIN TEMPORARILY NOT ALLOWED will be output.

When the break-in has been effected the break-in job obeys ENDJOB and the control job will request further commands in BREAKIN context.

Ending a MOP session

In order to finish a MOP session the operator should LOGOUT his MOP job if applicable, then ABANDON the control job.

Library entrants accessed

The job description for MOP on operator's console is held in :MACROS.OPMOPJD. This job description loads the operator's MOP subject program from :LIB.PROGRAM XKLY.

The break-in job description is held in :MACROS. OPSBIJD, and this loads a subject program :LIB.PROGRAM XKLZ.

Notes

- 1 For every operator's MOP job running there is also a background job. If break-in is required a further background job must be run, and the control job temporarily becomes a MOP job. This proliferation of jobs may require some adjustment to the installation parameters MOPLIMIT, BACKGROUND and JOBLIMIT.
- 2 The operator's task may be made simpler by providing a macro to cut down the drudgery of typing the control job's jobname and username every time a QUESTION is ANSWERed.
- 3 The control job may be run from either central or remote nominated consoles; more than one such job may be run simultaneously provided that jobname/username combinations obey the usual rules of uniqueness.
- 4 Although no restrictions are placed on the usernames for the control job and break-in jobs each installation will no doubt adopt its own conventions. The username(s) chosen must have EXECUTE access to the macros and programs mentioned above. If the control job is run under a different user from the MOP job, then the operator must be aware of the control job user's password if he wishes to break in on the MOP job. If the control job is run under a low security user then the password parameter should be null. (If the password parameter is null it may be omitted from the RUNJOB command).
- 5 There is no double break-in facility to throw away unwanted output, and such output may be queued up very quickly by GEORGE, effectively blocking the console, so great care should be taken not to issue commands which generate a large amount of output in the MOP jobs monitoring file unless the job's report mask has been set to eliminate the output of such messages to the "MOP" (operator's) console.

NOTE ON USER NOTICE (8) TO GEORGE 3 & 4 OPERATING

Amendment 1

User Notice 8 to the manual *GEORGE 3 and 4 Operating*, in describing Amendment 1, states that "This amendment is being issued in the form of a reprint of the manual incorporating the amended text." It should be noted, however, that this "amended reprint" of the manual is being distributed as if it were a straightforward amendment and users will not therefore be charged for it.



L¹

PUBLICATION (NOTICE NO.)

30/1/74

4334

GEORGE 3 AND 4 OPERATION MANAGEMENT (20)

4310 ✓

GEORGE 3 AND 4 OPERATING (10) ✓

File one copy of this
notice with each of the
publications indicated.

DUAL ACCESS EDS 60

This User Notice describes the operating procedure for dual Access EDS 60 under GEORGE 3 and 4 Mark 7.

GLOSSARY

Definitions

<i>Dual Access or Twinned</i>	- a device with two sets of controls which enable it to be simultaneously operated by two device controllers.
<i>Spindle</i>	- a single drive unit of the Dual Access EDS 60 device.
<i>Cripple mode</i>	- the state of the configuration when one of the processors is not running under GEORGE 3 or 4 and/or the MPB (7210) is inoperable.

Relationships between processors and spindles

The absolute unit numbers of the dual access spindles must be the same in both processors. The relationship between a processor and a particular spindle can be grouped into one of three software categories. These are:

1 Shared

A shared spindle is one in which the dual-access capability is in use. Only user exofiles are allowed to access a spindle with this property.

2 Dedicated

A dedicated spindle is one which is controlled by only one of the processors and appears as being offline to the other processor. A spindle with this property may be put to any use and may contain the GEORGE chapter file, filestore, swap residences or user exofiles.

3 Undedicated

The reverse of dedicated. An undedicated spindle appears as being offline to the processor concerned and is dedicated to the other processor.

The relationship can be declared before GEORGE is loaded and have the following restrictions:

- 1 A spindle marked as shared in one processor must be marked as shared in the other processor.
- 2 A spindle marked as dedicated or undedicated in one processor must be marked as the reverse in the other processor.

DECLARING THE STATUS OF TWINNED SPINDLES

There are two ways in which this can be achieved; either by use of the DEDICATE and UNDEDICATE Gin mend macros or by similar directives to the GEORGE Loader/Allocator program. Unit numbers of twinned spindles not specified will be classed as shared (see TWIN command for changing the status of spindles while GEORGE is running).

The Gin mend macros

DEDICATE %A, %B, %C.....%Z

%A, %B, etc. are the operators' numbers of the twinned spindles to be dedicated. The macro notified GEORGE of the operators' numbers of the twinned spindles that are to be the sole use of the processor in which the restored version of GEORGE is to run. A corresponding UNDEDICATE macro must be present in the GIN restore for the other processor specifying the same spindles as the corresponding DEDICATE macro.

UNDEDICATE %A, %B, %C.....%Z

Indicates to GEORGE the operators' numbers of the twinned spindles dedicated to the other processor.

When a DEDICATE or UNDEDICATE macro is used the previous settings will not be remembered, that is, to dedicate units 56 and 57 the following card should be inserted in the restore pack:

DEDICATE 56, 57

The cards:

DEDICATE 56
DEDICATE 57

will result in only unit 57 being dedicated.

If either DEDICATE or UNDEDICATE are issued without parameters all previous settings will be lost.

ERROR CHECKS

When GEORGE is loaded the parameters to the DEDICATE and UNDEDICATE Gin mend macros are checked:

- 1 If any of the units specified do not refer to a dual access spindle the message

DEDICATE/UNDEDICATE MACROS:UNITS*numbers* DO NOT REFER TO TWINNED SPINDLES.

- 2 If any operators number is specified more than once in either or both macros:

DEDICATE/UNDEDICATE MACROS:UNITS *numbers* HAVE BEEN SPECIFIED MORE THAN ONCE.

If either of these messages are output the system will enter the postmortem routine with the message:

PLEASE CORRECT DEDICATE/UNDEDICATE MACROS

The Loader/Allocator

The Loader/Allocator program will accept the directives DC and UN and these have the same specification as the Gin Mend macros DEDICATE and UNDEDICATE.

Example

An installation has a 7-spindle dual access EDS 60 with unit numbers 50 to 56. Units 50 and 51 are to be used for filestore residences in processor A. Processor B requires unit 54 for the GEORGE chapter file and a swap residence. The Gin restores for the two processors are as follows:

A

XKOV pack
#LIS 3
#COR 180
DEDICATE 50,51
UNDEDICATE 54
#LIS-1
#DEL

B

XKOF pack
#LIS 3
#COR 180
DEDICATE 54
UNDEDICATE 50, 51
#LIS-1
#DEL

Units 52, 53, 55 and 56 will be shared.

Using the Loader/Allocator to obtain the same effect:

A

Load George EXEC.
FI *nn* GEO3SUBFILE
LOADED
DC 50, 51
UN 54
DA *123456 GEORGE FILE
LOADED
RESTORE?

B

Load George EXEC.
FI *nn* GEO3SUBFILE
LOADED
DC 54
UN 50, 51
DA *765432 GEORGE FILE
LOADED
RESTORE?

FILESTORE RESIDENCES AND THE GEORGE CHAPTER RESIDENCE

Filestore residences

If a filestore residence is to be held on a dual access EDS 60 spindle the spindle must be dedicated to the processor concerned before GEORGE is loaded.

GEORGE chapter residence

If the GEORGE chapter residence is to be placed on a dual access EDS 60 spindle the spindle concerned must be dedicated, as for filestore residences.

The following restrictions are imposed if the GEORGE chapter residence is placed on a dedicated dual access EDS 60 spindle:

- 1 the exofiles holding the GEORGE chapter residences must have different names in both processors,

or

- 2 the storage unit serial number of the storage unit holding the GEORGE chapter residence must be specified in the direct access directive when loading GEORGE.

USE OF THE IPB 7210

Exofile twinning will open two IPC links through the IPB in opposite directions.

The link descriptions will be:

*identity*₁ TWDA

*identity*₂ TWDA

where *identity*₁ and *identity*₂ are the eight character

names assigned to the processors concerned using the IDENTITY Gin Macro. The identities of the processors must be different.

GEORGE will use the IPB as if it were an IPC user and in no way restricts the use of the device to itself unless this is required. (See Gin Macro IPCLH.)

OPERATOR COMMANDS

The TWIN command (TN):

TWIN *directive* U *number*

Forbidden context

Not operator.

Character strings

The first parameter is a character string defining the function to be applied to the number parameter. Each character string may be contracted, but the first two characters must be present. In some cases the number parameter is not required. The character strings are:

INITIALISE
UP or DOWN
SHARE, DEDICATE, UNDEDICATE

INITIALISE

This directive must be issued in both processors (though not necessarily simultaneously) and allows access to any shared spindles.

The number parameter must be present and specifies the operators number of the IPB channel through which the twinning dialogue is to be passed.

On receipt of the command an attempt is made to open IPC links for reading and writing.

SUCCESSFUL ESTABLISHMENT OF LINKS

If the IPC links are established an initialising message containing details of the status of each twinned spindle with regard to each processor is sent and checks are made according to the following table:

TABLE 1

		<u>Processor 1</u>		
		<i>Dedicated</i>	<i>Shared</i>	<i>Undedicated</i>
Processor 2	<i>Dedicated</i>	x	x	OK
	<i>Shared</i>	x	OK	OK*
	<i>Undedicated</i>	OK	OK*	OK*

If a box is marked:

X GEORGE will enter the postmortem routine with the message:

TWINNED UNITS *number(s)* HAVE CONFLICTING STATUS

OK* Both processors will have the relevant spindle marked as shared.

OK The status of the spindle with regard to both processors is satisfactory.

On successful completion of the command messages T3⁺ to T5⁺ will be output as appropriate followed by message T1⁺. ⁺ Messages referred to appear on page 8.

UNSUCCESSFUL ESTABLISHMENT OF LINKS

This condition can arise for several reasons:

- 1 Inoperable IPB
- 2 Other processor is broken or not running under GEORGE 3/4 Mark 7.
- 3 Other processor is running under GEORGE but has not issued the TWIN command with the initialise directive.

In all the above cases message T6 will be output to the operators console and action should be taken according to the following table:

TABLE 2

		<i>Action</i>	<i>Event</i>	<i>Comment</i>
1	IPB Inoperable	Issue the TWIN command with UP directive	Messages T3 to T5 and T2 are output to console	No access to shared spindles permitted.
2	Other processor broken or not running under GEORGE	Issue the TWIN command with DOWN directive	Messages as for successful establishment	Full access to shared spindles permitted
3	No TWIN command with INITIALISE directive issued in other processor	Issue the TWIN command with the INITIALISE directive in other processor	Messages as for successful establishment	Full access to shared spindles permitted

Notes:

Case 1 When the IPB becomes operable the TWIN command with the initialise directive need not be reissued since the IPC links will be automatically established.

Case 2 When the TWIN command with the initialise directive is issued in the other processor the establishment of the IPC links is automatic and hence the TWIN/initialise command need not be reissued in the originally "working" processor.

If the other processor is running under an operating system other than GEORGE 3/4 Mark 7 when the TWIN/down command was issued both processors may have uncontrolled access to the twinned spindles.

CHANGING THE IPB CHANNEL USED FOR TWINNING DIALOGUE

If it is required to use another IPB channel for twinning dialogue the TWIN/initialise command can be reissued at any time. The existing IPC links will be closed and the links re-established through the channel specified.

The messages T3 to T5 and T7 will be output to the operators' console followed by message T6.

ERROR MESSAGES ASSOCIATED WITH INITIALISE DIRECTIVE

UNIT *z* IS NOT AN IPB DEVICE

THE PROCESSORS CONNECTED BY UNIT *z* HAVE THE SAME IDENTITY

THERE ARE NO TWINNED UDAS DEVICES ON THIS INSTALLATION

TWIN COMMAND IN PROGRESS

INFORMATORY MESSAGE ASSOCIATED WITH INITIALISE DIRECTIVE

T1 EXOFILE TWINNING SYSTEM USING IPB UNIT *z*

T2 TWINNING SYSTEM NOT INITIALISED - PROCESSOR LINK FAIL

T3 UNITS *z* ARE SHARED

T4 UNITS *z* ARE DEDICATED

T5 UNITS *z* ARE UNDEDICATED

T6 PLEASE ISSUE TWIN COMMAND GIVING STATE OF OTHER PROCESSOR

T7 TWINNING LINKS CLOSED AND REOPENED ON IPB *z*

UP and DOWN

These two directives (in which the *number* parameter is not required) enable the operator to inform GEORGE of the state of the GEORGE in the other processor. This is necessary since it is not possible for GEORGE to distinguish between IPB inoperable and GEORGE in the other processor not working.

These directives must only be issued in response to message T6.

UP

If this directive is issued as a parameter to the TWIN command GEORGE will assume the IPB to be inoperable and no further access will be allowed to the system control area of the storage units on shared spindles. Exofiles already open remain open and transfer peris are still

permitted unless access to the flaw area is required.

DOWN

This directive should be issued under the following conditions:

- 1 The other processor has broken.
- 2 The other processor is not running under GEORGE or has not reached "GEORGE Ready".

After issuing this directive the processor concerned takes control of all the shared spindles and no attempt is made to communicate with the other processor regarding exofile usage.

Under no circumstances should TWIN/DOWN be issued such that it is operative in both processors simultaneously.

ERROR MESSAGES ASSOCIATED WITH UP AND DOWN DIRECTIVES:

GEORGE HAS NOT REQUESTED THIS DIRECTIVE TO BE ISSUED.

DEDICATE, UNDEDICATE, SHARE

This group of directives is used to change the status of a twinned spindle with regard to a processor.

In all cases the number parameter is required and must specify the operators number of a twinned spindle.

The effect of issuing these directives will only last for the duration of the GEORGE run in which they were issued, that is, if GEORGE breaks the status of the twinned spindle at the next "GEORGE Ready" control will revert to the spindles specified in the DEDICATE/UNDEDICATE Gin mend macros or Loader Allocator directives.

DEDICATE

This directive causes the processor in which it is issued to have full access to the storage unit on the specified spindle. This directive can only be issued under *cripple mode* conditions, therefore the status of the spindle with regard to the other processor is unaffected.

On completion of the command the message:

UNIT z IS DEDICATED

is output on the operators' console.

UNDEDICATE

This directive causes the spindle specified to appear as being offline to the processor in which it is issued. This directive can only be issued under *cripple mode* conditions and therefore the status of the spindle is not changed in the other processor.

On completion of the command the message:

UNIT *z* IS UNDEDICATED

is output on the operators' console.

SHARE

This directive causes the specified spindle to be marked as shared and can be issued under the following conditions:

- 1 If the processor in which it is issued has passed "GEORGE Ready" but has not issued TWIN/Initialise. In this case the spindle specified will only be marked as shared in the issuing processor.
- 2 If both processors are running under GEORGE and the TWIN/Initialise commands have been issued so that the processors are connected, both processors will mark the specified spindle as shared. If the command is issued under these conditions it need only be issued from one of the processors.

On completion of the command the message:

UNIT *z* IS SHARED

is output on the operators' console.

	IPB inop (TWIN UP issued)	Other processor down (TWIN DOWN issued)	ALL OK	To "GEORGE Ready" but no TWIN initialise yet issued
Dedicate	Dedicated	Dedicated	Error (E4)	Dedicated
Undedicate	Undedicated	Undedicated	Error (e4)	Undedicated
Share	Error (E5)	Shared	Shared in both	Shared

Table summarising the effect of the TWIN command with DEDICATE/UNDEDICATE/SHARE directives. The effect stated is that obtained in the issuing processor unless stated otherwise.

Notes

- 1 When the IPB becomes operable or the other processor comes alive checks will be made according to Table 1 (page 6).
- 2 When dedicating a spindle when the IPB is inoperable the undedicate directive should be issued in one processor before issuing the dedicate directive in the other.

ERROR MESSAGES ASSOCIATED WITH DEDICATE/UNDEDICATE/SHARE

- E1 UNIT *z* IS NOT A TWINNED UDAS DEVICE
- E2 TWIN COMMAND IN PROGRESS
- E3 UNIT *z* IS IN USE
- E4 THIS COMMAND IS ONLY ALLOWED IF THE IPB OR THE OTHER PROCESSOR IS DOWN
- E5 UNIT *z* CANNOT BE SHARED DUE TO PROCESSOR LINK FAILURE

CHANGES TO EXISTING OPERATOR COMMANDS

WHATPER

The WHATPER command will output information regarding the status of the dual access spindles. For example:

UNIT *z* IDLE (SHARED)

UNIT *z* ENGAGED (DEDICATED)

REDON (for non-standard discs)

The following message will be output on the operators console if the storage unit is not on a dedicated spindle:

UNIT *z* IS NOT DEDICATED

CARTRIDGE

The following error message will be output on the operators' console if the unit to which the cartridge is to be moved is not dedicated:

SPINDLES MUST BE DEDICATED

EFFECT ON JOB DESCRIPTIONS AND PROGRAMS

Access to exofiles on a shared spindle

The normal integrity check applies and the maximum access to a particular exofile will therefore be one writer and any number of readers simultaneously taken over by both processors.

Job descriptions and programs should specify the storage unit serial number whenever possible as this will reduce the number of IPB transfers required to open an exofile.

S-trusted programs

S-trusted programs will not be permitted to open the System Control Area files, that is, ICLSCAFILE or SYFLE, if the associated storage unit is not on a spindle dedicated to the processor being used. An attempt to open a system file for writing while the spindle is shared will result in error message M3 (see below) being output to the job's monitoring file.

Access to the System Control Area of a storage unit

In order to safeguard the System Control Area of a storage unit on a shared spindle only one processor is allowed to access the system information at a time. Therefore when executing an ONLINE command or peri that required use of the System Control Area the job may be held waiting with the message M2 while the right to access the system information is being obtained from the other processor. The user may break in at this point. Transfer peris are not affected by this possible delay except when access to the flaw index or flaw area is required.

Failure of processor link

If the IPB connecting the processors fails one of the messages M1 or M4 will be output to the monitoring file depending on the stage reached in executing the ONLINE command or peri instruction.

Transfer peris are not affected by processor link failure unless the flaw area or flaw index has to be accessed when message M5 will be output to the monitoring file.

In all cases the peri instruction will be failed after the appropriate message has been output.

Monitoring file messages

M1 STORAGE UNIT z NOT SEARCHED DUE TO FAILURE OF PROCESSOR LINK

M2 WAITING TO ACCESS SCA OF STORAGE UNIT z

- M3 FILE *filename* CANNOT BE OPENED BECAUSE YOU ARE USING AN S-TRUSTED PROGRAM AND THE STORAGE UNIT IS NOT ON A SPINDLE THAT IS DEDICATED TO THIS PROCESSOR
- M4 FILE *filename* CANNOT BE OPENED DUE TO PROCESSOR LINK FAILURE
- M5 NO FLAW ACTION ON STORAGE UNIT *z* BECAUSE OF PROCESSOR LINK FAIL

OPERATOR ACTION UNDER CRIPPLE MODE CONDITION

When a cripple mode condition arises the message:

PLEASE ISSUE TWIN COMMAND GIVING STATE OF OTHER PROCESSOR

is output to the operators' console.

During TWIN/Initialise

If cripple mode occurs during TWIN/Initialise action should be taken according to Table 2 on page 7.

During normal running

If cripple mode occurs when the system had previously been running satisfactorily the following action should be taken depending on the cause of the cripple mode condition:

1 Failure of the other processor

The TWIN/Down command should be issued in the live processor. If the other processor has entered the GEORGE postmortem routines this fact will automatically be notified to the live processor and the TWIN/Down command will be automatically issued by GEORGE in the live processor.

2 Inoperability of the IPB link

The TWIN/Up command should be issued in both processors or the TWIN/Initialise command can be reissued specifying a new IPB channel to be used.

If the TWIN/Up directive is issued no new access to the shared spindles will be allowed unless the TWIN/Dedicate/Undedicate commands are issued to change the status of the shared spindles with regard to the processors.

SYSTEM VULNERABILITY

Under certain circumstances the security of data on the storage units is not fully under the control of GEORGE. These are:

- 1 If an operating system other than GEORGE 3/4 Mark 7 is in use in both processors.
- 2 If GEORGE 3/4 Mark 7 is in use in one processor and another operating system is in use in the other.

The most probable combination here would be operators' executive in one processor and GEORGE 3/4 Mark 7 in the other. When using operators' executive care should be taken that no access is made to the EDS 60 units.

GEORGE should be restored to disc under GEORGE and the Loader/Allocator used to load the system (see also *GEORGE chapter residence*, page 4).

- 3 If the IPB link breaks down.

The inoperability of the IPB does not directly cause the possibility of data corruption but GEORGE relies on the operator for information regarding the state of the system which cannot be verified, that is, link or processor fail?

If the IPB is inoperable the operator has the task of distributing the shared spindles between the processors. He must ensure that when a particular spindle is dedicated to a processor, the other processor holds the opposite property regarding that spindle, that is, that the spindle is undedicated to the other processor.

MOVEMENT OF STORAGE UNITS

At all times

When a storage unit is to be taken off line it should always be done by pressing the hold button. If a storage unit is to be moved from a twinned spindle to a spindle that is not twinned or vice versa it should be ensured that no exofiles are open by issuing a WHATPER command for that spindle.

In the event of a storage unit being moved from a shared spindle to a dedicated spindle and GEORGE discovers that exofiles are still open when the unit is engaged on the new spindle the message:

FILE(S) OPEN ON STORAGE UNIT z WHEN ON A SHARED
DEVICE

will be output on the operators' console and the storage

unit disengaged. The storage unit should be replaced on a shared spindle.

During IPB inoperability

Under normal conditions the processors decide which is to disengage on receipt of a hold button interrupt on a shared spindle. If the IPB is inoperable the spindle cannot be disengaged by GEORGE but the message:

UNIT z CANNOT BE DISENGAGED : PROCESSOR LINK
FAILED - PLEASE DISENGAGE

will be output on the console typewriter of both processors. At this stage the stop button may be used to take the storage unit down.

ACTION OVER SYSTEM BREAKS

ZPMCORE command and postmortems

If one processor breaks and the error is thought likely to be in the IPC or Twinning routines a ZPMCORE command should be in the line processor as soon as possible after the break. The command issued from the operators' console should be:

ZC IPC,TWN,TRA

If the ZPMCORE command is not available a normal GEORGE Postmortem should be taken in the line processor as soon as possible after the other processor breaks.

A specification of the ZPMCORE command is attached.

Status of spindles

If the TWIN command with the SHARE, DEDICATE or UNDEDICATE parameter was issued during the run of GEORGE the settings will be lost over a system break. (See page). It is therefore necessary to check that the status of the dual access spindles with regard to each processor comply with Table 1, page 6 before the TWIN/Initialise command is issued.

ZPMCORE (ZC)

Function

Causes specified information to be collected from GEORGE's core while GEORGE is running; the information will be written to the System Journal and a job will be initiated to list the information collected. The files :JOURNAL.COREPRINT and :LIB.PROGRAM XK75 must be input to the filestore.

Format

ZPMCORE parameter list

There are currently five parameter types, but more may be supplied by the user, provided he writes his own information collection routine. The interface for adding parameters and mending in collection routines is described above.

The following are currently available built-in parameter types.

ADR(n_1, n_2)

Where n_1, n_2 are numeric strings optionally terminated by 1 or more characters C.

n_1 defines a start address in core from which data is to be collected. The use of C indicates, that the address required is 'replaced', that is, the relevant address is contained in the location specified. The address may be replaced to any degree. n_2 defines the length of the area of core required to be collected; like n_1 this length may be replaced any number of times.

Examples ADR(20,40)
 specified the area 40 words long starting at
 word 20
 ADR(150CC, 50C)
 specifies the area starting at the address
 contained in the word whose address is
 contained in word 150, and the length of
 which is contained in word 50.

DEV(n_1, n_2)

Where n_1, n_2 , etc. [to a maximum of 10 parameters] are operators' numbers of peripherals whose device records and APIAs are required to be collected and output. As with ADR the parameters n_1, n_2 , etc. may be replaced, so that any number of characters C may optionally follow a parameter to indicate that the geographical number will be found in a certain location.

Example DEV(50,51)
 specifies that device records and additional
 peripheral information areas for units 50
 and 51 are to be collected.

IPC

Means collect diagnostic information from core relevant to Inter Program Communication.

TWN

Means collect diagnostic information from core relevant to twinned devices.

TRA

Means collect information from the trace buffer.

Forbidden contexts

NOT OPERATOR

Execution

For each main parameter GEORGE will collect from core the areas specified (the collection taking place without coordination) and write them to the system journal in the form of unpacked records with message number 500. When all parameters have been dealt with a job is initiated (running under :JOURNAL) which will search the system journal and list (in a rather crude format) the information collected.

The amount of information collected is currently limited to 493 words for each contiguous area of core specified (implicitly or explicitly). Any area specified which exceeds this limit will be truncated to the first 493 words and an explanatory message output.

Errors in parameters will be indicated by appropriate messages, and the next parameter will be dealt with; in the case of DEV, however, an error in a geographical number will lead to the next number in the list being dealt with.

Error messages

There are two types of error messages. The first type are messages arising from errors during typing the command (for example, misspelling and general syntax errors). These are output to the command source. The second type concerns errors which are normally discovered during execution. The latter are output to System Journal and will be printed with the output from the listing program.

SYNTAX ERRORS

- 1 ERROR IN %B IN ZPMCORE: %C FORMAT ERROR
 - if no components given for DEV or ADR parameters.
- 2 ERROR IN %B IN ZPMCORE: PARAMETER FORMAT ERROR %C
 - if components of DEV or ADR in form other than that of a number optionally followed by any number of C characters.
- 3 ERROR IN %B IN ZPMCORE: %C NOT RECOGNISED.
 - if parameter specified other than those recognised.

- 4 ERROR IN %B IN ZPMCORE: RESERVATION VIOLATION
- if address of the area required is outside
GEORGE's core (both DEV and ADR)
- 5 ERROR IN DEV IN ZPMCORE; %C IS AN ILLEGAL
GEOGRAPHICAL NUMBER - if %C is not a valid
geographical number.
- 6 ERROR IN ZPMCORE: NO PARAMETERS
- if no parameters specified. Command error
action ensures.

SEMANTIC ERRORS

- 1 POINTER TO NEXT AREA REQUIRED IS INVALID
- if address of device information is outside of
GEORGE's core.
- 2 THE LENGTH OF FOLLOWING AREA TRUNCATED TO 493 WORDS
- if the length is calculated as being more than
493 words.
- 3 ONLY FIRST 10 DEVICES OF DEV PARAMETER WILL BE
CONSIDERED
- if more than 10 components specified in DEV
parameter.
- 4 NEGATIVE OR ZERO LENGTH SPECIFIED
- if length of the area required is calculated as
negative or zero.

© International Computers Limited, Reading 1974



L.

PUBLICATION (NOTICE NO.)

24/7/74

4310

GEORGE 3 AND 4 OPERATING (11)

File one copy of this
notice with each of the
publications indicated.

AMENDMENTS TO THE MANUAL

Please note the following alterations to the manual
due to the release of GEORGE 3 and 4 Mark 8.10.

Page 27, Line 5

Insert at the end of this line:

"In the case of central line printers, the headings
and terminators may be augmented by the printing of
character arrays designed to aid operational debatching
and despatching (see the documentation of the macro
LFDEBATCHERS in Appendix 1 of the ICL manual *GEORGE 3
and 4 Operation Management*, TP 4334)."

Page 27, Line 23

Insert after this line:

"Heading formats, as described above, may be modified
for local line printers by the restore time macros
LFDEBATCHERS and LFENDTHROW. These macros are
documented in Appendix 1 of the ICL manual *GEORGE 3 and
4 Operation Management* TP 4334."

Page 42.4, Lines 25 to 27

Replace these lines with:

"If the listing is to be redirected to the CENTRAL
cluster, the property CENTRAL must be specified unless
the CHANGELIST command is issued on the CENTRAL
operators' console."

Replace these lines with:

Function

- 1 Changes the properties specified in a LISTFILE entry in :SYSTEM.OUTPUT to those specified as a parameter to this command. If no property parameter is specified, the entry is modified to require the console property attributed to the peripheral on which the command was issued (if no such attribution exists, CENTRAL is assumed).

While this command will operate on any properties, its major use is expected to be the alteration of console properties in order to change the routing of LISTFILE.
- 2 Changes the urgency information in an entry in :SYSTEM.OUTPUT to that specified as a parameter to this command. The entries are ordered according to urgency thus a change in urgency will change the order, influencing the sequence in which files are listed.

Note that functions 1 and 2 cannot be mixed in a single use of the command.

Format

- 1 CHANGELIST *user name, job name, output peripheral type, file description, PROPERTY property names*
- 2(a) CHANGELIST *user name, job name, output peripheral type, file description, URGENCY a*
- 2(b) CHANGELIST *user name, job name, URGENCY a*

Forbidden contexts

NOT OPERATOR

Execution

FORMATS 1 AND 2(a)

The appropriate entry or entries are found in :SYSTEM.OUTPUT using the first four parameters, plus any console property attributed to the device on which the command was issued. The action taken by the command on the entries it finds depends upon the format as follows:

- 1 The properties specified in a property parameter will be used to replace any properties in the entry

in :SYSTEM.OUTPUT. If no property parameter is specified, any properties in the entry will be replaced by the console property attributed to the peripheral on which the command was issued. If no such attribution exists CENTRAL is assumed.

- 2(a) The entry in :SYSTEM.OUTPUT will be changed by replacing its urgency by that specified in the fifth parameter and will then be appropriately re-ordered in :SYSTEM.OUTPUT.

If no urgency parameter is present function 1 is assumed.

FORMAT 2(b)

All entries in :SYSTEM.OUTPUT which were produced by the job specified by the first two parameters will have their urgency changed to that specified in the third parameter and will then be appropriately re-ordered in :SYSTEM.OUTPUT.

If no urgency parameter is present a command error is generated.

Page 87, Line -11

Insert after this line, before "Error messages" the following sentence:

"If the command is issued from a remote nominated console, the only entries to be changed are those destined for a device in that console's cluster".

Page 87, Line -1

Add to the list of error messages:

"PARAMETER FORMAT ERROR: Z IS OUT OF RANGE"

Page 145, Line 3

Delete from this line the phrase:

"and NO USER activities"

Page 145, Lines 23, 24

Delete from these lines the phrase:

"or NO USER activity"

Page 146, Line 10

The information in the column headed "*Information given*" should now read:

"The user name and job name"

Page 154, Lines 2, 3

Delete fro these lines the phrase:

'and NO USER context activity"

Page 154, Line 19

Replace this line with:

"If Format 3 is used, information is output about every job."

© International Computers Limited, Reading 1974



Library

PUBLICATION (NOTICE NO.)

13/11/74

4310

GEORGE 3 AND 4 OPERATING (12)

File one copy of this
notice with each of the
publications indicated.

AMENDMENTS TO THE MANUAL

Page 22.5, Line 2

Insert the following new section after this line:

"Store impediment messages

It can happen in GEORGE 3 that the swapping in of an object program is impeded by the presence of a locked block in store. This is known as a store impediment. One of two messages is produced to inform the operators that such a situation has arisen. If a peripheral transfer takes much longer than expected, the message:

STORE IMPEDIMENT: UNTERMINATED TRANSFER ON UNIT U_n

will be produced where n is the offending unit's number. Users should normally report occurrences of the message to their site engineer as a possible hardware fault, and not submit error reports. If the message is repeated frequently it is possible that the condition may be cured by engineering attention to the specified peripheral.

If the store impediment is not associated with an unterminated transfer of the message:

STORE IMPEDIMENT: *string*

will be produced, where *string* is a series of up to three octal numbers. The occurrence of this message should be reported to ICL via the normal channels, as it indicates the presence of a bug. If this form of the message does occur it should only be transient and will not warrant breaking the system.

Page 60, Lines 17 to 22

Replace these lines with:

"The postmortem routine will dump to any magnetic tape. If the tape is labelled GEORGEPM TAPE it will first be positioned before the trailer label, unless otherwise indicated by console input messages (see below), and then the dump will be given. If it is labelled SCRATCH TAPE, or if it has any other label and the SCR parameter is used (see below),"

Page 61, Line 2

Replace this line with:

"*tape serial number and deck number* can be followed by the character strings BOT or SCR, for example:"

Page 61, Line 3

Insert after this line:

"*tape serial number, deck number, SCR*

Page 61, Lines 5 to 7

Replace these lines with:

" *UT tape serial number, deck number, SCR*

The BOT character string will cause the postmortem dump to be written to the tape immediately after the header label and, in effect, erases all previous postmortem dumps written to the tape. The SCR character string will cause the specified tape to be renamed GEORGEPM TAPE (irrespective of its previous name) and write the post-mortem dump to the tape immediately after the header label."

Page 165, Line 13

Insert the following two messages after this line:

STORE IMPEDIMENT: UNTERMINATED *Store impediment messages,*
TRANSFER ON UNIT Un *page 22.5*

STORE IMPEDIMENT: *string* *Store impediment messages,*
 page 22.5

© International Computers Limited, Reading 1974

To: Users' Group representatives.
Computing Service Staff.

Mistakes in ICL manuals

If you come across an error in an ICL manual, the department to notify at ICL is:

Software Distribution,
30/31 Friar Street,
Reading. Telephone: Reading 581258

They will either deal with it themselves (if it concerns the assembly and distribution of manuals or user notices) or they will pass it on to Data Processing Publications Department (who write ICL manuals) or to the ICL Printing Services at Letchworth (who print and bind them).

Do not expect a swift response, or any response at all! With manuals, the stock answer is "will be corrected in the next amendment". Amendments to all manuals are (in theory) regularly prepared by ICL, but have been known to be delayed.



Library

PUBLICATION (NOTICE NO.)

8/1/75

4334 GEORGE 3 AND 4 OPERATION MANAGEMENT (31)

4310 GEORGE 3 AND 4 OPERATING (13)

File one copy of this
notice with each of the
publications indicated.

This User Notice replaces GEORGE 3 and 4 Operation
Management (20) and GEORGE 3 and 4 Operating (10).

DUAL ACCESS EDS 60

This User Notice describes the operating procedure for
dual access EDS 60 under GEORGE 3 and 4.

GLOSSARY

Definitions

<i>Dual access or Twinned device</i>	- a device with two sets of controls which enable it to be simultaneously operated by two device controllers.
<i>Spindle</i>	- a single drive unit of the dual access EDS 60 device.
<i>Cripple mode</i>	- the state of the configuration when one of the processors is not running under GEORGE 3 or 4 and/or the IPB (7210) is inoperable.

Relationships between processors and spindles

The absolute unit numbers of the dual access spindles
must be the same in both processors. The relationship
between a processor and a particular spindle can be
grouped into one of three software categories. These
are:

1 Shared

A shared spindle is one in which the dual access
capability is in use. A spindle with this
property can only contain user exofiles.

2 Dedicated

A dedicated spindle is one which is controlled by only one of the processors and appears as being offline to the other processor. A spindle with this property may be put to any use and may contain the GEORGE chapter file, filestore, swap residences or user exofiles.

3 Undedicated

The reverse of dedicated. An undedicated spindle appears as being offline to the processor concerned and is dedicated to the other processor.

The relationship can be declared before GEORGE is loaded and has the following restrictions:

- 1 A spindle marked as shared in one processor must be marked as shared in the other processor.
- 2 A spindle marked as dedicated or undedicated in one processor must be marked as the reverse in the other processor.

DECLARING THE STATUS OF TWINNED SPINDLES

There are two ways in which this can be achieved; either by use of the DEDICATE and UNDEDICATE Gin macros or by similar directives to the GEORGE Allocator/Loader program. Unit numbers of twinned spindles not specified will be classed as shared (see TWIN command for changing the status of spindles while GEORGE is running).

The Gin macros

DEDICATE $n_1, n_2, n_3 \dots n_{24}$

n_1, n_2 , etc. are the operators' numbers of the twinned spindles to be dedicated: up to 24 parameters can be specified provided that the total length of the source line does not exceed 72 characters. The macro notifies GEORGE of the operators' numbers of the twinned spindles that are to be the sole use of the processor in which the restored version of GEORGE is to run. A corresponding UNDEDICATE macro must be present in the GIN restore for the other processor specifying the same spindles as the corresponding DEDICATE macro.

UNDEDICATE $n_1, n_2, n_3 \dots n_{24}$

Indicates to GEORGE the operators' numbers of the twinned spindles dedicated to the other processor.

When a DEDICATE or UNDEDICATE macro is used the previous settings will not be remembered, that is, to dedicate units 56 and 57 the following card should be inserted in the restore pack:

DEDICATE 56, 57

The cards:

DEDICATE 56
DEDICATE 57

will result in only unit 57 being dedicated.

If either DEDICATE or UNDEDICATE are issued without parameters all previous settings will be lost and all twinned spindles will be marked as shared.

ERROR CHECKS

When GEORGE is loaded the parameters to the DEDICATE and UNDEDICATE Gin macros are checked:

- 1 If any of the units specified do not refer to a dual access spindle the message

DEDICATE/UNDEDICATE MACROS:UNITS*numbers* DO NOT REFER TO TWINNED SPINDLES.

- 2 If any operators number is specified more than once in either or both macros:

DEDICATE/UNDEDICATE MACROS:UNITS *numbers* HAVE BEEN SPECIFIED MORE THAN ONCE.

If either of these messages are output the system will enter the postmortem routine with the message:

PLEASE CORRECT DEDICATE/UNDEDICATE MACROS

The Allocator/Loader

The Allocator/Loader program will accept the directives DC and UN and these have the same specification as the Gin macros DEDICATE and UNDEDICATE.

Example

An installation has a 7-spindle dual access EDS 60 with unit numbers 50 to 56. Units 50 and 51 are to be used for filestore residences in processor A. Processor B requires unit 54 for the GEORGE chapter file and a swap residence. The Gin restores for the two processors are as follows:

A

XKOF pack
#LIS 3
#COR 180
 DEDICATE 50,51
 UNDEDICATE 54
#LIS-1
#DEL

B

XKOF pack
#LIS 3
#COR 180
 DEDICATE 54
 UNDEDICATE 50, 51
#LIS-1
#DEL

Units 52, 53, 55 and 56 will be shared.

Using the Allocator/Loader to obtain the same effect:

<i>A</i>	<i>B</i>
Load George EXEC.	Load George EXEC.
FI nn GEO3SUBFILE	FI nn GEO3SUBFILE
LOADED	LOADED
DC 50, 51	DC 54
UN 54	UN 50, 51
DA *123456 GEORGE FILE	DA *765432 GEORGE FILE
LOADED	LOADED
RESTORE?	RESTORE?

FILESTORE RESIDENCES AND THE GEORGE CHAPTER RESIDENCE

Filestore residences

If a filestore residence is to be held on a dual access EDS 60 spindle the spindle must be dedicated to the processor concerned before GEORGE is loaded.

GEORGE chapter residence

If the GEORGE chapter residence is to be placed on a dual access EDS 60 spindle the spindle concerned must be dedicated, as for filestore residences.

The following restrictions are imposed if the GEORGE chapter residence is placed on a dedicated dual access EDS 60 spindle:

- 1 the exofiles holding the GEORGE chapter residences must have different names in both processors,
- or
- 2 the storage unit serial number of the storage unit holding the GEORGE chapter residence must be specified in the direct access directive when loading GEORGE.

USE OF THE IPB 7210

Exofile twinning will open two IPC links through the IPB in opposite directions.

The link descriptions will be:

*identity*₁ TWDA

*identity*₂ TWDA

where *identity*₁ and *identity*₂ are the eight character names assigned to the processors concerned using the

IDENTITY Gin Macro (see page 234 of the manual *GEORGE 3 and 4 Operation Management*). The identities of the processors must be different.

GEORGE will use the IPB as if it were an IPC user and in no way restricts the use of the device to itself unless this is required (see Gin Macro IPCLIM, page 236 of the manual *GEORGE 3 and 4 Operation Management*).

OPERATOR COMMANDS

The TWIN command (TN):

TWIN directive, Unnumber

Forbidden context

Not operator.

Character strings

The first parameter is a character string defining the function to be applied to the number parameter. Each character string may be contracted, but the first two characters must be present. In some cases the number parameter is not required. The character strings are:

INITIALISE
UP or DOWN
SHARE, DEDICATE, UNDEDICATE

INITIALISE

This directive must be issued in both processors (though not necessarily simultaneously) and allows access to any shared spindles.

The number parameter must be present and specifies the operators number of the IPB channel through which the twinning dialogue is to be passed.

On receipt of the command an attempt is made to open IPC links for reading and writing.

SUCCESSFUL ESTABLISHMENT OF LINKS

If the IPC links are established an initialising message containing details of the status of each twinned spindle with regard to each processor is sent and checks are made according to the following table:

TABLE 1

		<u>Processor 1</u>		
		<i>Dedicated</i>	<i>Shared</i>	<i>Undedicated</i>
<u>Processor 2</u>	<i>Dedicated</i>	x	x	OK
	<i>Shared</i>	x	OK	OK*
	<i>Undedicated</i>	OK	OK*	OK*

If a box is marked:

x GEORGE will enter the postmortem routine with the message:

TWINNED UNITS *number(s)* HAVE CONFLICTING STATUS

OK* Both processors will have the relevant spindle marked as shared.

OK The status of the spindle with regard to both processors is satisfactory.

On successful completion of the command messages T3⁺ to T5⁺ will be output as appropriate followed by message T1⁺. Messages referred to appear on page 8.

UNSUCCESSFUL ESTABLISHMENT OF LINKS

This condition can arise for several reasons:

- 1 Inoperable IPB
- 2 Other processor is broken or not running under GEORGE 3/4.
- 3 Other processor is running under GEORGE but has not issued the TWIN command with the initialise directive.

In all the above cases message T6 will be output to the operators console and action should be taken according to the following table:

TABLE 2

	<i>Action</i>	<i>Event</i>	<i>Comment</i>
1 IPB Inoperable	Issue the TWIN command with UP directive	Messages T3 to T5 and T2 are output to console	No access to shared spindles permitted
2 Other processor broken or not running under GEORGE	Issue the TWIN command with DOWN directive	Messages as for successful establishment	Full access to shared spindles permitted
3 No TWIN command with INITIALISE directive issued in other processor	Issue the TWIN command with the INITIALISE directive in other processor	Messages as for successful establishment	Full access to shared spindles permitted

Notes:

Case 1 The action stated is for use when the IPB cannot be made operable immediately. If the IPB is merely off-line it should be online at this stage. When the IPB becomes operable the TWIN command with the initialise directive need not be reissued since the IPC links will be automatically established.

Case 2 When the TWIN command with the initialise directive is issued in the other processor the establishment of the IPC links is automatic and hence the TWIN/initialise command need not be reissued in the originally "working" processor.

If the other processor is running under an operating system other than GEORGE 3/4 Mark 7 when the TWIN/down command was issued both processors may have uncontrolled access to the twinned spindles.

CHANGING THE IPB CHANNEL USED FOR TWINNING DIALOGUE

If it is required to use another IPB channel for twinning dialogue the TWIN/initialise command can be reissued at any time. The existing IPC links will be closed and the links re-established through the channel specified.

The messages T3 to T5 and T7 will be output to the operators' console followed by message T6 if the IPC links cannot be established through the new channel.

ERROR MESSAGES ASSOCIATED WITH INITIALISE DIRECTIVE

UNIT *z* IS NOT AN IPB DEVICE

THE PROCESSORS CONNECTED BY UNIT *z* HAVE THE SAME IDENTITY

THERE ARE NO TWINNED UDAS DEVICES ON THIS INSTALLATION

TWIN COMMAND IN PROGRESS

Note: the "TWIN COMMAND IN PROGRESS" error message will be output if a TWIN/initialise command has previously been issued and has not terminated, that is, messages T1 and T2 have not been output to the operator's console at the time of issuing another TWIN/initialise command.

INFORMATORY MESSAGE ASSOCIATED WITH INITIALISE DIRECTIVE

T1 EXOFILE TWINNING SYSTEM USING IPB UNIT *z*

T2 TWINNING SYSTEM NOT INITIALISED - PROCESSOR LINK FAIL

T3 UNITS *z* ARE SHARED

T4 UNITS *z* ARE DEDICATED

T5 UNITS *z* ARE UNDEDICATED

T6 PLEASE ISSUE TWIN COMMAND GIVING STATE OF OTHER PROCESSOR

T7 TWINNING LINKS CLOSED AND REOPENED ON IPB *z*

UP and DOWN

These two directives (in which the *number* parameter is not required) enable the operator to inform GEORGE of the state of the GEORGE in the other processor. This is necessary since it is not possible for GEORGE to distinguish between IPB inoperable and GEORGE in the other processor not working.

These directives must only be issued in response to message T6 and can therefore only be used under *cripple mode* conditions. The effect of these directives is cancelled by GEORGE when normal mode conditions are restored. If neither of these directives are issued in response to message T6 any jobs requiring use of shared spindles will be held waiting until normal mode conditions are restored.

UP

If this directive is issued as a parameter to the TWIN command GEORGE will assume the IPB to be inoperable and no further access will be allowed to the system control area of the storage units on shared spindles, (see page 13 under heading "Failure of processor link"). Exofiles already open remain open and transfer peris are still permitted unless access to the flaw area is required.

DOWN

This directive should be issued under the following conditions:

- 1 The other processor has broken.
- 2 The other processor is not running under GEORGE or has not reached "GEORGE Ready".

After issuing this directive the processor concerned takes control of all the shared spindles and no attempt is made to communicate with the other processor regarding exofile usage.

Under no circumstances should TWIN/DOWN be issued such that it is operative in both processors simultaneously.

ERROR MESSAGES ASSOCIATED WITH UP AND DOWN DIRECTIVES:

GEORGE HAS NOT REQUESTED THIS DIRECTIVE TO BE ISSUED.

DEDICATE, UNDEDICATE, SHARE

This group of directives is used to change the status of a twinned spindle with regard to a processor.

In all cases the number parameter is required and must specify the operators number of a twinned spindle.

The effect of issuing these directives will only last for the duration of the GEORGE run in which they were issued, that is, if GEORGE breaks the status of the twinned spindles at the next "GEORGE Ready" will revert to that specified in the DEDICATE/UNDEDICATE Gin macros or Allocator/Loader directives.

DEDICATE

This directive causes the processor in which it is issued to have full access to the storage unit on the specified spindle. This directive can only be issued under *cripple mode* conditions, therefore the status of the spindle with regard to the other processor is unaffected.

Under no circumstances should the TWIN/dedicate command be issued so that the same spindle is dedicated to both processors (see Note 2, page 11).

The GIN Macro DECOM can be used to inhibit use of this command and therefore prevents a spindle being dedicated to both processors (see page 22).

On completion of the command the message:

UNIT *z* IS DEDICATED

is output on the operators' console.

UNDEDICATE

This directive causes the spindle specified to appear as being offline to the processor in which it is issued. This directive can only be issued under *cripple mode* conditions and therefore the status of the spindle is not changed in the other processor.

On completion of the command the message:

UNIT *z* IS UNDEDICATED

is output on the operators' console.

SHARE

This directive causes the specified spindle to be marked as shared and can be issued under the following conditions:

- 1 If the processor in which it is issued has passed "GEORGE Ready" but has not issued TWIN/Initialise. In this case the spindle specified will only be marked as shared in the issuing processor.
- 2 If both processors are running under GEORGE and the TWIN/Initialise commands have been issued so that the processors are connected, both processors will mark the specified spindle as shared. If the command is issued under these conditions it need only be issued from one of the processors.

On completion of the command the message:

UNIT z IS SHARED

is output on the operators' console.

	IPB inop (TWIN UP issued)	Other processor down (TWIN DOWN issued)	All OK	To "GEORGE Ready" but no TWIN initialise yet issued
Dedicate	Dedicated	Dedicated	Error (E4)	Dedicated
Undedicate	Undedicated	Undedicated	Error (E4)	Undedicated
Share	Error (E5)	Shared	Shared in both	Shared

Table summarising the effect of the TWIN command with DEDICATE/UNDEDICATE/SHARE directives. The effect stated is that obtained in the issuing processor unless stated otherwise.

Notes:

- 1 When the IPB becomes operable or the other processor comes alive checks will be made according to Table 1 (page 6).
- 2 When dedicating a spindle when the IPB is inoperable the undedicate directive should be issued in one processor before issuing the dedicate directive in the other.

ERROR MESSAGES ASSOCIATED WITH DEDICATE/UNDEDICATE/SHARE

- E1 UNIT z IS NOT A TWINNED UDAS DEVICE
- E2 TWIN COMMAND IN PROGRESS
- E3 UNIT z IS IN USE
- E4 THIS COMMAND IS ONLY ALLOWED IF THE IPB OR THE OTHER PROCESSOR IS DOWN
- E5 UNIT z CANNOT BE SHARED DUE TO PROCESSOR LINK FAILURE

Notes:

- 1 Error message E2 will be output if a previous TWIN/initialise or TWIN/share command has been issued, but has not yet been terminated.
- 2 Error message E3 will be output if exofiles are open on the unit specified by the parameter.

CHANGES TO EXISTING OPERATOR COMMANDS

WHATPER

The WHATPER command will output information regarding the status of the dual access spindles. For example:

```
UNIT z IDLE ( SHARED )  
UNIT z ENGAGED ( DEDICATED )
```

REDON (for non-standard discs)

The following message will be output on the operators console if the storage unit is not on a dedicated spindle:

```
UNIT z IS NOT DEDICATED
```

CARTRIDGE

The following error message will be output on the operators' console if the unit to which the cartridge is to be moved is not dedicated:

```
SPINDLES MUST BE DEDICATED
```

EFFECT ON JOB DESCRIPTIONS AND PROGRAMS

Access to exofiles on a shared spindle

The normal integrity check applies and the maximum access to a particular exofile will therefore be one writer and any number of readers simultaneously taken over both processors.

Job descriptions and programs should specify the storage unit serial number whenever possible as this will reduce the number of IPB transfers required to open an exofile.

S-trusted programs

S-trusted programs will not be permitted to open the System Control Area files, that is, ICLSCAFILE or SYFLE, if the associated storage unit is not on a spindle dedicated to the processor being used. An attempt to open a system file for writing while the spindle is shared will result in command error C1 if the ONLINE command is in use or comment M3 in the case of an unanticipated peri.

Access to the System Control Area of a storage unit

In order to safeguard the System Control Area of a storage unit on a shared spindle only one processor is allowed to access the system information at a time.

Therefore when executing an ONLINE command or peri that requires use of the System Control Area the job may be held waiting with the message M2 while permission to access the system information is being obtained from the other processor. The user may break in at this point. Transfer peris are not affected by this possible delay except when access to the flaw index or flaw area is required.

Failure of processor link

If the IPB connecting the processors fails, that is the TWIN UP command has been issued, action taken depends on whether an ONLINE command or unanticipated peri was in use. For the ONLINE command the comment M1 followed by command error C3 or command error C2 only, will be output to the jobs monitoring file depending on the stage reached in executing the ONLINE command. For unanticipated peris comments M1 or M4 will be output to the jobs monitoring file and the program is restarted with an integrity error or file not found reply set in the peri control area.

Transfer peris are not affected by processor link failure unless the flaw area or flaw index has to be accessed when message M5 will be output to the monitoring file, and the peri instruction is failed.

Monitoring file messages (Comment Category)

- M1 STORAGE UNIT *z* NOT SEARCHED DUE TO FAILURE OF PROCESSOR LINK
- M2 WAITING TO ACCESS SCA OF STORAGE UNIT *z*
- M3 FILE *filename* CANNOT BE OPENED BECAUSE YOU ARE USING AN S-TRUSTED PROGRAM AND THE STORAGE UNIT IS NOT ON A SPINDLE THAT IS DEDICATED TO THIS PROCESSOR
- M4 FILE *filename* CANNOT BE OPENED DUE TO PROCESSOR LINK FAILURE
- M5 NO FLAW ACTION ON STORAGE UNIT *z* BECAUSE OF PROCESSOR LINK FAIL

Command errors associated with the ONLINE command

- C1 ERROR IN ONLINE : FILE *filename* CANNOT BE OPENED BECAUSE YOU ARE USING AN S-TRUSTED PROGRAM AND THE STORAGE UNIT IS NOT ON A SPINDLE THAT IS DEDICATED TO THIS PROCESSOR.
- C2 ERROR IN ONLINE : FILE *filename* CANNOT BE OPENED DUE TO PROCESSOR LINK FAILURE
- C3 ERROR IN ONLINE : FILE *filename* NOT FOUND

OPERATOR ACTION UNDER CRIPPLE MODE CONDITION

When a cripple mode condition arises the message:

PLEASE ISSUE TWIN COMMAND GIVING STATE OF OTHER
PROCESSOR

is output to the operators' console.

During TWIN/Initialise

If cripple mode occurs during TWIN/Initialise action should be taken according to Table 2 on page 7.

During normal running

If cripple mode occurs when the system had previously been running satisfactorily the following action should be taken depending on the cause of the cripple mode condition:

1 Failure of the other processor

The TWIN/Down command should be issued in the live processor. If the other processor has entered the GEORGE postmortem routines this fact will automatically be notified to the live processor and the TWIN/Down command will be automatically issued by GEORGE in the live processor.

2 Inoperability of the IPB link

The TWIN/Up command can be issued in both processors or the TWIN/Initialise command can be reissued specifying a new IPB channel to be used.

If the TWIN/Up directive is issued jobs that are currently obeying an ONLINE *DA command or a non-transfer peri instruction will be failed (see page 13). No new access to the shared spindles will be allowed unless the TWIN/Dedicate/Undedicate commands are issued to change the status of the shared spindles with regard to the processors.

SYSTEM VULNERABILITY

Under certain circumstances the security of data on the storage units is not fully under the control of GEORGE. These are:

- 1 If an operating system other than GEORGE 3/4 is in use in both processors.
- 2 If GEORGE 3/4 is in use in one processor and another operating system is in use in the other.

The most probable combination here would be operators' executive in one processor and GEORGE 3/4 in the other. When using operators' executive, care should be taken that no access is made to the twinned EDS 60 units.

If GEORGE is to be held on disc, it should be restored under GEORGE and the Allocator/Loader used to load the system (see also *GEORGE chapter residence*, page 4).

3 If the IPB link breaks down.

The inoperability of the IPB does not directly cause the possibility of data corruption but GEORGE relies on the operator for information regarding the state of the system which cannot be verified, that is, link or processor fail?

If the IPB is inoperable the operator has the task of distributing the shared spindles between the processors. He must ensure that when a particular spindle is dedicated to a processor, the other processor holds the opposite property regarding that spindle, that is, that the spindle is undedicated to the other processor.

MOVEMENT OF STORAGE UNITS

At all times

When a storage unit is to be taken off line it should always be done by pressing the hold button. If a storage unit is to be moved from a shared spindle to a spindle that is not shared or vice versa it should be ensured that no exofiles are open by issuing a WHATPER command for that spindle.

In the event of a storage unit being moved from a shared spindle to a dedicated spindle and GEORGE discovers that exofiles are still open when the unit is engaged on the new spindle the message:

FILE(S) OPEN ON STORAGE UNIT *z* WHEN ON A SHARED
DEVICE

will be output on the operators' console and the storage unit disengaged. The storage unit should be replaced on a shared spindle.

During IPB inoperability

Under normal conditions the processors decide which is to disengage on receipt of a hold button interrupt on a shared spindle. If the IPB is inoperable the spindle cannot be disengaged by GEORGE but the message:

UNIT 2 CANNOT BE DISENGAGED : PROCESSOR LINK
FAILED - PLEASE DISENGAGE

will be output on the console typewriter of both processors. At this stage the stop button may be used to take the storage unit down.

ACTION OVER SYSTEM BREAKS

Postmortems

If one processor breaks and the error is thought likely to be in the IPC or Twinning routines a postmortem should be taken in the live processor as soon as possible after the other processor breaks.

If it is inconvenient to take such postmortems in the live processor, further assistance in obtaining evidence can be obtained from ICL. A diagnostic facility known as ZPMCORE is available for Dual Access EDS 60 and IPC users to employ under ICL supervision. (The description of ZPMCORE published in the previous version of this User Notice should be ignored. In particular, certain aspects which were inadvertently published there have not been fully implemented.)

Status of spindles

If the TWIN command with the SHARE, DEDICATE or UNDEDICATE parameter was issued during the run of GEORGE the settings will be lost over a system break. (See page 9). It is therefore necessary to check that the status of the dual access spindles with regard to each processor comply with Table 1, page 6 before the TWIN/Initialise command is issued.

Example: Use of the TWIN commands

Configuration: IPB 7210 - Unit 62 3 x EDS60 Dual Access - Units 44, 45, 46

<i>IPB Online?</i>	<i>Processor A</i>	<i>Processor B</i>	<i>Comments</i>
Yes	Load George Executive FI nn GEO3SUBFILE LOADED DC 44 UN 46 DA GEORGE FILEA RESTORE? N GEORGE READY PLEASE INITIALISE EXOFILE TWINNING TN IN,62	Load George Executive FI nn GEO3SUBFILE LOADED DC 46 UN 44 DA GEORGE FILEB RESTORE? Y .	Unit 44 dedicated to A Unit 46 dedicated to B Unit 45 is shared User job in A requires access to the shared spindle (Unit 45). No access to shared spindles is allowed until the TN/initialise command is issued.

IPB. Online ?	Processor A	Processor B	Comments
Yes	PLEASE ISSUE TWIN COMMAND.. .. GIVING STATE OF OTHER PROCESSOR TN DOWN UNIT 44 DEDICATED UNIT 45 SHARED UNIT 46 UNDEDICATED EXOFIL TWINNING USING IPB UNIT 62 	 GEORGE READY TN IN, 62 UNIT 44 UNDEDICATED UNIT 45 SHARED UNIT 46 DEDICATED EXOFIL TWINNING USING IPB UNIT 62	Although the IPB is online, processor A cannot establish IPC links because the TN/ initialise command has not been issued in B As B has not yet reached GEORGE READY User jobs in A can now access shared spindles Open IPC links in Processor B IPC links are estab- lished in both process- ors

<i>IPB Online?</i>	<i>Processor A</i>	<i>Processor B</i>	<i>Comments</i>
NO	PLEASE ISSUE TWIN COMMAND GIVING STATE OF OTHER PROCESSOR TN UP WP 45 UNIT 45 ENGAGED (SHARED) SUSN 654321 TN DE, 45 UNIT 45 DEDICATED .	PLEASE ISSUE TWIN COMMAND GIVING STATE OF OTHER PROCESSOR TN UP WP 45 UNIT 45 ENGAGED (SHARED) SUSN 654321 TN UN, 45 UNIT 45 UNDEDICATED .	IPB goes offline. IPC links remain open but disestablished Both processors are told that the other processor is OK. Any jobs opening exofiles on shared discs will be failed, but exofiles already open remain open. IPC links open but disestablished. It is required to dedicate Unit 45 to A. First check no exofiles open by issuing WP command. Dedicate and undedicate using Twin commands
YES			IPB back online. IPC links are automatically re-established Required to share unit 46.

IPB Online?	Processor A	Processor B	Comments
	WP 46 UNIT 46 IDLE (UNDEDICATED) TN SH, 46 UNIT 46 SHARED . This processor breaks down . . . Postmortem being taken . . . DA GEORGE FILEA RESTORE? N GEORGE READY WP 44 WP 45 WP 46 UNIT 44 DEDICATED UNIT 45 SHARED UNIT 46 UNDEDICATED	WP 46 UNIT 46 IDLE (DEDICATED) UNIT 46 SHARED . PLEASE ISSUE TWIN COMMAND.. ..GIVING STATE OF OTHER PROCESSOR TN DO WP 44 WP 45 WP 46 UNIT 44 UNDEDICATED UNIT 45 UNDEDICATED UNIT 46 SHARED	Issue WP command to check if exofiles open then issue TWIN/share command in A or B, but not both as IPC links are established. Processor A breaks down. B is told of this using TN DOWN. Processor B has access to all shared spindles. IPC links are open but disestablished in B. At this stage the status of the twinned spindles in A reverts to that specified when using the Allocator/Loader (Page 3). These differ from B and can

<i>IPB Online?</i>	<i>Processor A</i>	<i>Processor B</i>	<i>Comments</i>
	TN DE, 45 UNIT 45 DEDICATED . TN SH, 46 UNIT 46 SHARED TN IN, 62 UNITS 44, 45 ARE DEDICATED UNIT 46 SHARED UNIT 44 SHARED TN SH, 45 UNIT 45 SHARED . . .	 TN SH, 44 UNIT 44 SHARED UNIT 45 SHARED . . .	be altered by issuing TWIN SH/DE/UN in A or B but only affects the processor in which they are issued. (Because IPC links are not yet established) Now establish the IPC links Required to share units 44, 45. The TWIN/SHARE command can be issued in either processor; the unit is marked 'shared' in both as IPC links are established. End of example.

The Gin Macro DECOM

Format:

DECOM *string*

where *string* can be the character strings ON or OFF.

Function:

Controls the use of the TWIN/dedicate command.

Context:

Anywhere that a #MEND is allowed.

Execution

OFF

Modifies GEORGE so that all uses of the TWIN/dedicate command result in the command error:

PARAMETER FORMAT ERROR

being output on the operators' console.

ON

Reverses the effect of the OFF parameter, that is the TWIN/dedicate command is implemented as normal.

Default

If this macro is not used the default is as if DECOM ON had been issued.

Error cases

If a parameter other than OFF or ON is specified the macro will have a null effect.

If no parameter is specified a GIN error will result with the message:

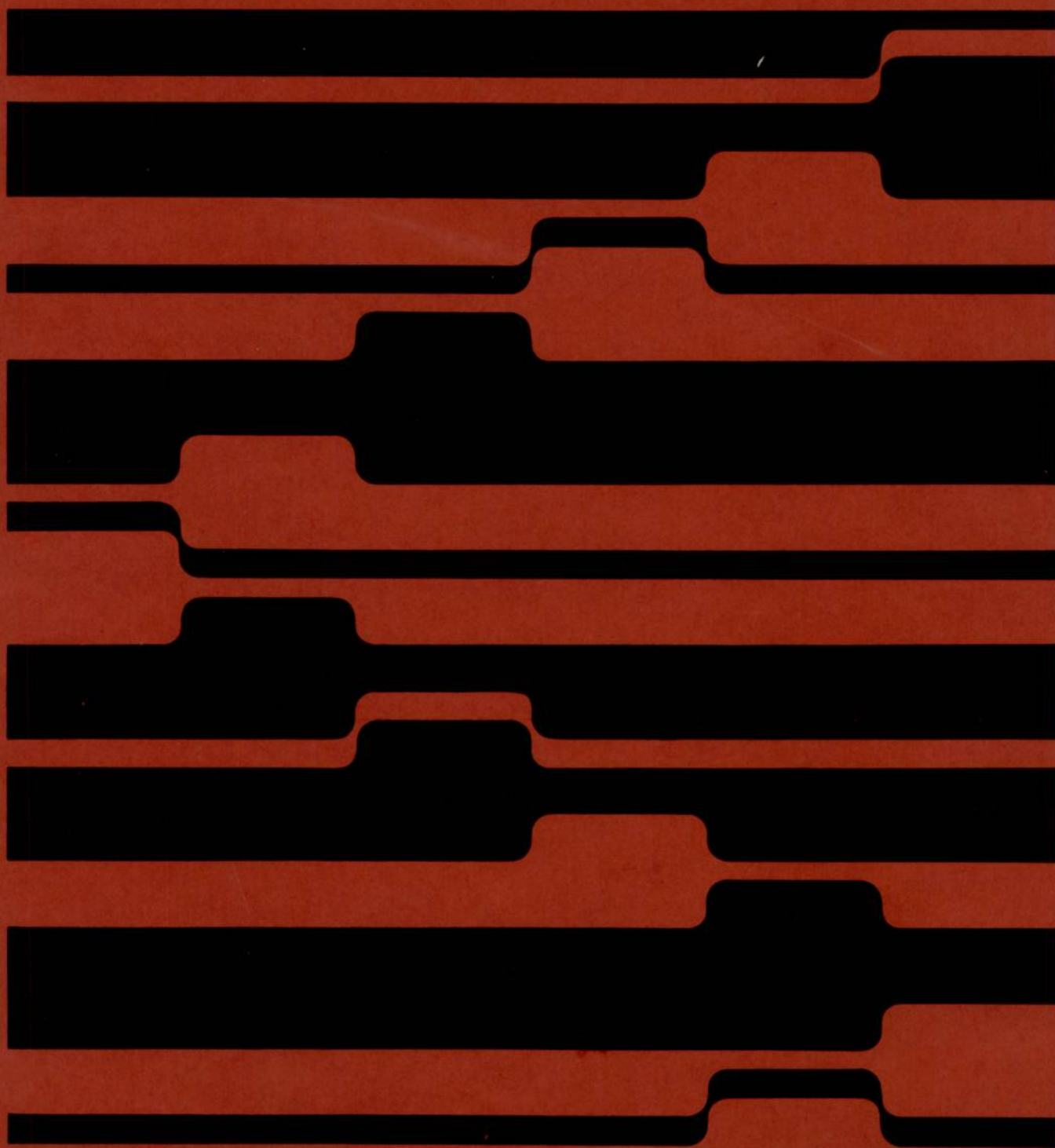
!!MUST HAVE ON OR OFF PARAMETER!!

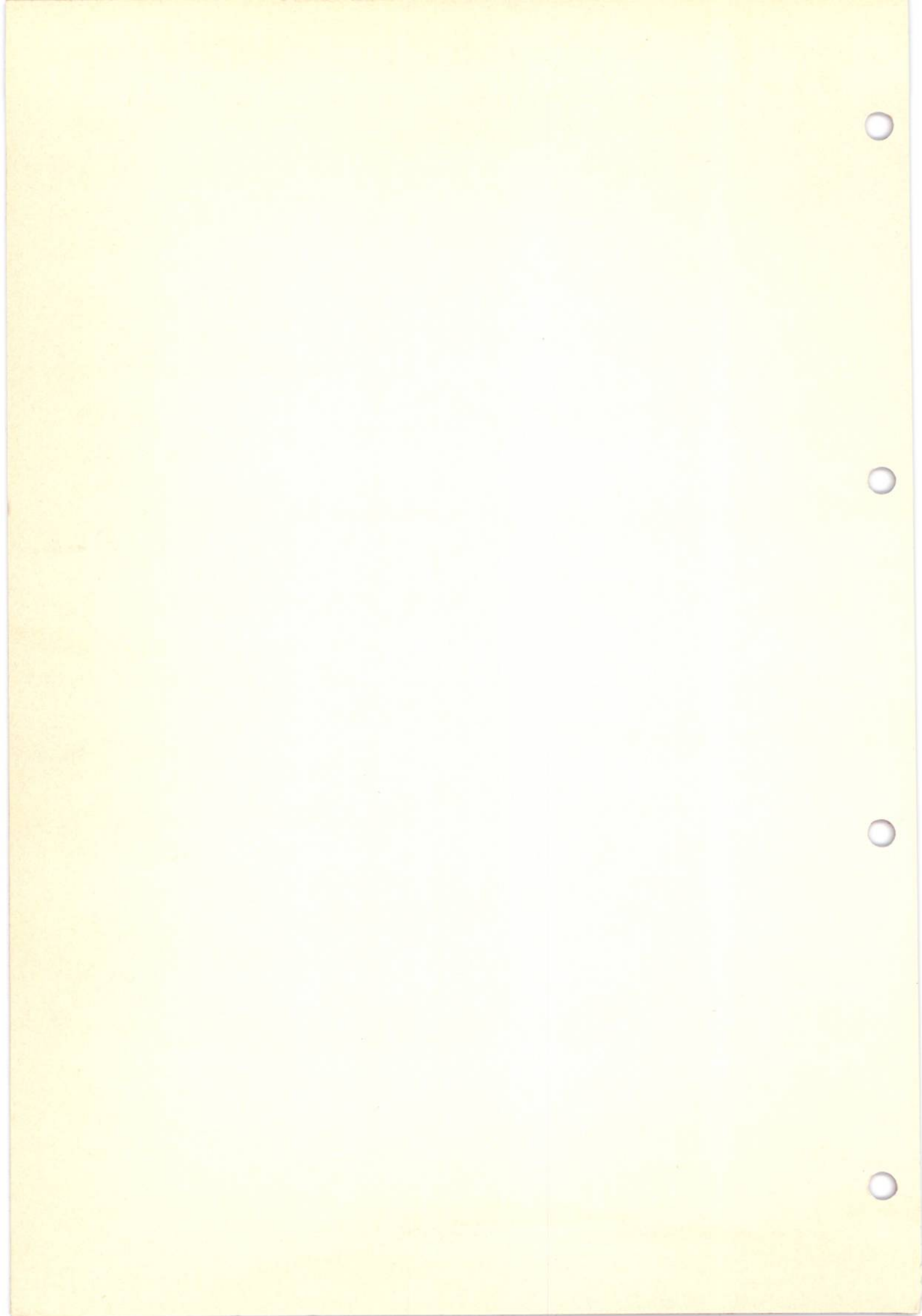
ICL

**GEORGE 3
and 4
Operating**

1900 Series

OXFORD UNIVERSITY COMPUTING LABORATORY
RECEIVED 20 MAR 1973
COMPUTING SERVICE 4310.





ICL

**GEORGE 3
and 4
Operating**

1900 Series



The policy of International Computers Limited is one of continuous development and improvement of its products and services, and the right is therefore reserved to alter the information contained in this document without notice. ICL makes every endeavour to ensure the accuracy of the contents of this document but does not accept liability for any error or omission. Any equipment or software performance figures and times stated herein are those which ICL expects to be achieved in normal circumstances. Wherever practicable, ICL is willing to verify upon request the accuracy of any specific matter contained in this document.

With effect from July 1968 the name of International Computers and Tabulators Limited has been changed to International Computers Limited

Technical Publication 4310

© International Computers Limited 1971

Third Edition October 1971

Reprinted January 1973, incorporating Amendment List 1

Issued by Technical Publications Service
International Computers Limited
Head Office: ICL House, Putney, London SW15
Produced by ICL Printing Services
at Letchworth, Hertfordshire

Preface

This manual is intended for the operator in an installation that is run under the control of the ICL 1900 Series operating systems GEORGE 3 and 4. In this edition *the current Mark* is Mark 7 GEORGE 3 which is intended for use with the 1902S and all larger unpagged machines in the Series except for those 1903, 1904, 1905 and 1909 processors whose number is not suffixed by a letter. GEORGE 4 offers similar facilities to GEORGE 3 but is intended for use with paged processors. Paging is a method of store utilization which helps to increase the efficiency of the system. In this manual the term GEORGE refers to features common to GEORGE 3 and GEORGE 4. Any facility unique to one system will be clearly specified as GEORGE 3 or GEORGE 4.

MINIMUM CONFIGURATIONS

Minimum configuration for GEORGE 3

The minimum configuration for GEORGE 3 is:

- a 1902S central processor with the following features:
 - extended mode
 - a real time clock
 - a program timer
 - the commercial computing feature
- 48K words of core store if programs are small; otherwise 64K is a reasonable minimum
- 2M words of direct access backing store (storage units and/or drum)
- four magnetic tape decks
- a line printer
- a card reader or paper tape reader

Minimum configuration for GEORGE 4

The minimum configuration for GEORGE 4 is:

- a 1903T central processor with the following features:
 - paging unit
 - extended mode
 - a real time clock
 - a program timer
 - the commercial computing feature
- 96K words of main store
- 8M words of direct access backing store
- 512K words of 2851 drum
 - four magnetic tape decks
 - two line printers
 - two card readers or paper tape readers

Minimum configuration for multiple on-line programming (MOP) (GEORGE 3 or 4)

If MOP is to be used, then 7071 teletypewriters or 7083/1 to 7086/1 termiprinters are required. These can be connected via:

- a 7007/2 multiplexer and 7008 data terminals
- a 7920 or 7930 scanner
- a 7900 communications processor and 7920 or 7930 scanner

Alternatively, if only a few terminals are being used, they can be connected via 7070 data terminals.

A reasonable configuration for MOP with ten to twenty terminals would include 96K words of core store, a drum and a storage unit.

Minimum configuration for Remote Job Entry (GEORGE 3 or 4)

If remote batch processing is to be used, then 7020 data terminals or 1901A processors on telephone lines are required. These can be connected via:

- 7010/3, 7010/5 or 7010/7 uniplexers
- a 7007/2 multiplexer and 7010/1, 7010/4 or 7010/6 data terminals
- a 7920 or 7930 scanner
- a 7900 communications processor and 7920 or 7930 scanner

Each 7020 terminal must have a 7023 teletypewriter connected to it.

CONTENTS OF THE MANUAL

Chapter 1 is an introduction that provides the operator with a brief description of operating systems in general and the GEORGE 3 and 4 operating systems in particular. The main functions of the operator in a GEORGE environment are listed and there is a description of the ways in which the operator can communicate with GEORGE and Executive.

Chapter 2 describes methods of setting up and loading GEORGE. Chapter 2.1 describes the part that the operator plays in processing the workload. Chapter 3 gives details of how the operator handles peripherals in a GEORGE environment. The back-up system is described in chapter 4 and the operation of the high level scheduler in Chapter 5. In Chapter 6 the operator is told how to deal with a breakdown of GEORGE and Chapter 7 describes the ways in which the operator can close down the system if necessary. All GEORGE commands and command macros that either can be issued by the operator or are of interest to him are fully described in alphabetical order in Chapter 8. The output the operator can receive from the system is listed in alphabetical order in Appendix 1. Appendix 2 gives details of the Geronimo system, which provides the operator with requested information on one or more local Mark II alphanumeric visual display units.

The following manuals are referred to in the text of this manual:

- 1 *Introduction to GEORGE 3* (Edition 2, TP 4265)
- 2 *Operating Systems GEORGE 3 and 4* (Edition 5, TP 4345)
- 3 *GEORGE 3 and 4 Operation Management* (Edition 4, TP 4334)

Contents

Preface	iii
MINIMUM CONFIGURATIONS	iii
Minimum configuration for GEORGE 3	iii
Minimum configuration for GEORGE 4	iii
Minimum configuration for multiple on-line programming (MOP)	iv
Minimum configuration for Remote Job Entry (RJE)	iv
CONTENTS OF THE MANUAL	iv
 Chapter 1 Introduction	 1
OPERATING SYSTEMS	1
GEORGE CHARACTERISTICS	1
GEORGE FACILITIES	1
Job description	1
The filestore	1
Off-lining	2
Editing	2
Privacy	2
Resilience	2
Control of peripherals	2
Remote batch processing	2
Multiple on-line programming	2
Scheduling	2
Accounting and budgeting	3
THE OPERATOR'S FUNCTION	3
OPERATOR COMMUNICATION WITH GEORGE AND EXECUTIVE	3
Operator communication with Executive	4
COMMUNICATION DURING LOADING OF GEORGE	4
COMMUNICATION DURING GEORGE BREAKDOWNS	4
Operator communication with GEORGE	4
INFORMATIVE OUTPUT	4
SYSTEM REQUESTS	4
OPERATOR COMMANDS	4
Cancelling characters on the console	5
GEORGE 3 error messages	5
 Chapter 2 Setting up GEORGE	 7
CONTENTS OF THE TAPE ICLKG3MARK_n (OR ICLKG4MARK_n)	7
WHAT SETTING UP GEORGE INVOLVES	7
SETTING UP GEORGE WITH AN OPERATOR'S EXECUTIVE	9

Restoring GEORGE to a direct access device	9
THE LOADER PROGRAM #XKOF	9
THE GIN PROGRAM	10
Loading the GEORGE Executive	11
Loading GEORGE	11
GELL	11
THE GEORGE LOADER	12
SETTING UP GEORGE WITH A GEORGE EXECUTIVE	13
Allocating the filestore residences, swap residences and the GEORGE chapter residence using the Allocator/Loader	13
Restoring and loading GEORGE using the Allocator/Loader	16
Allocator/Loader directive error messages	17
Mending GEORGE	21
THE LOADER PROGRAM #XKOF	21
THE GIN PROGRAM	22
EARLY MORNING START ROUTINE (E.M.S.)	22
 Chapter 2.1 Processing the workload	 22.3
JOBS	22.3
The WHATJOBS and WHATSTATE commands	22.3
THE OPERATOR'S FUNCTION	22.3
INSTALLATION PARAMETERS	22.3
Control of on-line backing store	22.4
Siting of object programs in core (GEORGE 3 only)	22.4
 Chapter 3 Peripheral handling	 23
INTRODUCTION	23
Jobs	23
On-line and off-line	23
BASIC PERIPHERALS	24
Input	24
JOB DESCRIPTIONS	24
OFF-LINE DATA	24
ON-LINE DATA	24
FAILURES DURING INPUT	25
OTHER PERIPHERAL EVENTS	25
UNLOADING BASIC INPUT PERIPHERALS	25
Output	26
MONITORING FILE	26
ENGAGING BASIC OUTPUT PERIPHERALS	26
HEADING FORMATS	27
FAILURES DURING OUTPUT	27
PAPERLOW	28
TERMINATION OF LISTINGS	28
OUTPUT TO SPECIAL STATIONERY	28
UNLOADING BASIC OUTPUT PERIPHERALS	28
Peripheral properties	29
PROPERTY NAME QUALIFIERS	29
CONSOLE PROPERTIES	30

ATTRIBUTING PROPERTIES	30
CANCELLING PROPERTIES AND ATTRIBUTIONS	30
EXAMPLES OF THE USE OF THE PROPERTY SYSTEM	30
Maintenance engineer	30
Using special stationery on line printers	31
Console properties and peripheral clusters	31
MAGNETIC TAPES	31
Insecure tapes	31
Secure tapes	32
POOL TAPES AND WORK TAPES	
Loading magnetic tapes	32
QUERY STATUS	34
MODE	34
Unloading magnetic tapes	35
Properties on magnetic tapes	35
ENTRANT PROPERTIES AND PERIPHERAL PROPERTIES	36
REFERRING TO PROPERTIES ON MAGNETIC TAPES	36
UNUSABLE PERIPHERALS	36
DIRECT ACCESS DEVICES	36
Loading direct access devices	36
Unloading direct access devices	36
Device failure messages	37
CONTROL AREA INVOLVED IN THE FAILING PERI ORDER	37
Exchanging filestore cartridges	39
COMMUNICATIONS SYSTEMS	39
Communications devices	39
CHARACTER BUFFERING DEVICES	40
MESSAGE BUFFERING DEVICES	40
Terminal devices	40
7020 REMOTE DATA TERMINALS	40
1901A REMOTE JOB ENTRY TERMINALS	40
TELETYPEWRITERS AND TERMIPRINTERS	40
System messages	41
Describing the communications configuration of an installation	41
PERIPHERAL IDENTIFICATION FROM OPERATOR'S CONSOLE	41
The Installation Description File (I.D.F.)	42
SETTING UP THE I.D.F.	42
ALTERING THE I.D.F. DURING RUNNING	42.1
LISTING THE I.D.F.	42.1
Conceptual multiplexers	42.1
WHAT IS A CONCEPTUAL?	42.1
SETTING UP CONCEPTUALS	42.1
The BROADCAST and SPEAK commands	42.2
Peripheral clusters	42.2
The use of 7020 remote data terminals	42.2

Operating clusters	42.2
OPERATING A REMOTE CONSOLE	42.3
OPERATING REMOTE BULK PERIPHERALS	42.3
REMOTE JOB ENTRY	42.3
RE-ROUTING LISTFILES	42.4
COMMANDS AVAILABLE TO THE REMOTE OPERATOR	42.4
INTER OPERATOR COMMUNICATION	42.5
Inoperable MOP terminals and jobs in limbo	42.5
LIMBO	42.5
EXOTIC PERIPHERALS	42.5
INTER PROGRAM COMMUNICATION (IPC)	42.6
Operational aspects of IPC	42.6
IPC and the property system	42.7
 Chapter 4 The back-up system	 43
THE START ROUTINES	43
Backmap	43
General restore routine	44
TAPE FAILS AND FORMAT ERRORS	44
Juggernaut restorer	45
The :MACROS.EMSJOBS facility	45
Dump checking	45
THE MAINTENANCE ROUTINES	46
The backing store unjammer	46
The incremental dumper	47
CANCELLING A DUMP	49
THE DUMP TOTAL FACILITY	49
DUMP TAPE RELEASING	49
The file retrieval system	49
DUMP TAPE PROCESSING	49
The dump tape processor	50
The tape-to-tape processor (TTTP)	51
THE OPERATION OF THE TAPE-TO-TAPE PROCESSOR	52
MESSAGES	54
 Chapter 5 Scheduling	 57
THE OPERATION OF THE HIGH LEVEL SCHEDULER	57
THE COMPUTING POWER INDEX (C.P.I.)	58
SAVING BACKGROUND JOBS	58
URGENCY LEVELS	58.1
THE WHATJOBS AND WHATSTATE COMMANDS	58.1
THE EXPRESS COMMAND	58.1
 Chapter 5.1 Accounting	 58.3
LOG ANALYSIS	58.3
The Journal Accounting Program	58.3
 Chapter 6 System breakdowns and reloading	 59
SYSTEM ERRORS	59
GEORGE first error	59
LINE PRINTER OUTPUT	60
POSTMORTEM DUMP TO MAGNETIC TAPE	60

CHANGING THE POSTMORTEM DUMP TAPE	60
ACTION AFTER POSTMORTEM COMPLETE	61
LISTING POSTMORTEM DUMPS FROM MAGNETIC TAPE	61
Exception conditions	62
GEORGE second error	62
Loading after system error	62
HARDWARE, OTHER SOFTWARE, OPERATOR OR SYSTEM LIMITATION ERRORS	62.1
OTHER SYSTEM BREAKDOWNS	62.1
 Chapter 7 Closing down the system	 63
OBJECT	63
METHOD OF CLOSEDOWN	63
Cancellation of FINISH	64
Important note	64
 Chapter 8 Specifications of GEORGE commands	 65
INTRODUCTION	65
Name	65
Function	65
Format	65
PARAMETER FORMATS	65
PARAMETER QUALIFIERS	68
Forbidden contexts	68
OPERATOR, NOT OPERATOR	68
READER, NOT READER	68
USER, NO USER	68
REMOTE, NOT REMOTE	68
Execution	68
Error messages	68
COMMON COMMAND ERRORS	68
Errors in verbs	69
Context errors	69
Errors in parameters	69
ABANDON (AB)	71
ANSWER (AN)	73
ASSOCIATE (AE)	75
ATTACH (AH)	78
ATTRIBUTE (AU)	81
BROADCAST (BC)	83
CANCEL (CC)	84
CANTDO (CD)	85
CARTRIDGE (CG)	86
CHANGELIST (CH)	87
CONCEPTUAL (CL)	88
DEAD (DD)	90
DISENGAGE (DG)	91
DOCUMENT (DOCU or DM)	92

DUMP (DU)	93
EXPRESS (EP)	95
FINISH (FN)	97
HARDWARE (HW)	100
IDFLIST	102
INPUT (IN)	103
INSTPARA (IP)	105
JOB (JB)	110
LABEL (LB)	112
LISTINCINDEX (LX)	113
LISTSERIAL (LS)	114
MODE (MO)	115
MOP (MP)	116
NEW (NE)	118
PAGES (PA)	120
PMPRINT	121
PROPERTY (PR)	123
REDON (RO)	125
RETRIEVE (RV)	129
RIGHT (RI)	131
RUNJOB (RJ)	132
SPEAK (SK)	133
STOPLIST (SL)	134
TAPERIGHT (TH)	136
TAPES (TP)	137
TAPEWRONG (TW)	138
TERMINATE (TE)	139
TRANSPORTS (TR)	140
UNLOAD (UL)	141
VIDEO	143
WHATJOBS (WJ)	145
WHATLIST (WL)	147
WHATPER (WP)	148
WHATPROPS (WA)	153
WHATSTATE (WS)	154
WRONG (WR)	156
 Appendix 1 Operator messages	 157
 Appendix 2 Geronimo	 169
INTRODUCTION	169
IMPLEMENTATION	169
System set-up	169
USING GERONIMO	170
Activation	170
Basic rules for using Geronimo	170
AUTO facility	170
Normal commands	171

The MASTER AVDU	171
MASTER COMMANDS	172
Status values	172
Close down	172
Remote Job Entry	172
USING THE RJE FACILITY	173
ERRORS AND DIAGNOSTICS	174
System-detected errors	174
GEORGE 3 and 4 detected errors	175
AVDU failure	175
Postmortem and diagnostic tracing facilities	175
 Index	 177

Chapter 1 Introduction

OPERATING SYSTEMS

Operating systems are designed to organize the flow of work through computers. The principal aim of such a system is to minimize the amount of work handled by the operator, (because operator intervention acts as a bottleneck in the flow of work), and hence to enable the computer itself to handle the organization of jobs. On the smaller machines in the ICL 1900 Series, the combination of Executive and the operator is adequate to control the flow of work but on larger machines the use of an operating system can substantially increase efficiency.

The advantages of an operating system are particularly significant in the context of a large organization which has access to powerful computer facilities, when it becomes necessary to regulate the activities of numerous branches within that organization. In such an environment where a large number of small programs are being processed, human intervention can absorb a significant proportion of machine time. By using the power of the computer to handle such tasks as setting up basic peripherals and communicating with Executive, the operating system is able to minimize the proportion of machine time taken up by operator action. An operating system also offers some degree of privacy in so far as it limits access to information and enables different branches to operate independently of each other. Various branches of an organization may require to have direct access to the computer by means of on-line terminals such as console typewriters. Each on-line user must be guaranteed a fast response, but Executive-controlled multiprogramming cannot achieve this. Therefore, an operating system is required to enable each user's needs to be satisfied.

An operating system can organize and perform centrally operations such as off-lining of data and dumping of data for reasons of resilience, thus encouraging efficient use of the machine. It can also organize and schedule the allocation of the machine's resources.

GEORGE CHARACTERISTICS

GEORGE has three characteristics that distinguish it from other operating systems:

- 1 It is a general operating system in that it is designed for all kinds of users. Almost all present and future 1900 Series software can be run under GEORGE and all languages used with the 1900 Series can be used on an installation that is run under GEORGE
- 2 It is an integrated system in that it can be used simultaneously for conventional processing and on-line work, generally offering the same facilities for both modes of operation.
- 3 It can be used on a wide range of configuration (see the Preface, page iii).

GEORGE 4 offers similar facilities to GEORGE 3 but is intended for use with paged processors. Paging is a method of store utilization which helps to increase the efficiency of the system. In this manual the term GEORGE refers to features common to GEORGE 3 and GEORGE 4. Any facility unique to one system will be clearly specified as GEORGE 3 or GEORGE 4.

GEORGE FACILITIES

Job descriptions

When the user wishes to submit a job for batch processing under GEORGE (known as a *background job*) he inputs a series of instructions to the operating system to control the running of the job. These instructions are written in the GEORGE *command language* and together constitute the *job description*. They replace the written instructions to the operator which the user would have to submit in an installation without an operating system.

The filestore

The provision of a filestore enables the user to off-line input and output for his object programs and to keep information permanently available for easy access. The filestore is held primarily on direct access devices, and

is organized as a hierarchical structure. Data in the filestore may be organized in the format of a basic file (suitable for inputting and outputting on basic peripherals), a magnetic tape file, or a direct access file, whichever is most convenient for the user. Files are *device-independent* in that the user refers to them by name and is not concerned with their physical position. Basic files are also device-independent as regards input medium; thus a card reading program can have its data punched on paper tape.

Off-lining

Data and programs can be input to the filestore at any time. When the user wishes to run a job he can issue commands in his job description telling GEORGE to read the input data from the filestore and write the output data to the filestore. The output files can be listed at any time, either by GEORGE command or by program. This is much more efficient than using peripherals on-line.

Editing

A built-in editor is provided for the user to edit any of his basic files in the filestore.

Privacy

A user is able to impose a security restriction on the running of jobs in his name by means of the SECURITY command and the password system, and a system of *traps* enables the user to control what kind of access, if any, other users may have to his files. To grant another user access to one of his files, the user must issue a command specifying the other user's name, the filename, and the type of access to be permitted.

Resilience

Some resilience in the event of a system break is provided in the form of a regular dumping system. At intervals specified by the installation manager, any files that have been altered or inserted since the last dump are copied to magnetic tape. Thus, if the direct access backing store is corrupted by the hardware (or by GEORGE), the user has a copy of his files as they were at the last dump.

Control of peripherals

Normally object programs will not directly control peripherals, but will be connected to filestore files instead. This frees programs from the limitations imposed by the number of peripherals available and the speed at which peripheral transfers can be performed. There may be cases, however, where the user must control a peripheral himself and he can issue commands in his job description to have peripherals allocated to his program for use on-line.

Remote batch processing

Groups of peripherals remote from the central computer room can be connected to 7020 data terminals or 1901A processors and linked to the central processor via multiplexers or uniplexers. Such groups can be defined to form *clusters*; mini-installations for bulk input and output and remote job entry.

Multiple on-line programming

The part of GEORGE that is concerned with multi-access is called MOP (Multiple On-line Programming). GEORGE allows a number of users to run interactive jobs whilst batch jobs are being run as background.

In the present version of GEORGE the devices that can be used for on-line work are 7071 console typewriters and 7083/1 to 7086/1 termiprinters. Later versions will cater for other devices.

MOP enables users to feed information into the system, to run programs and interact with them, and to receive output at the terminal. This mode of processing is particularly well suited to jobs that fall naturally into small sections between which human intervention is required; for example, program development, file enquiries or editing. However, it is also possible to initiate background jobs from a MOP terminal.

The MOP user communicates with GEORGE by inputting commands in the GEORGE command language. Whereas in a background job GEORGE stores the user's commands in a job description file before obeying them, in a MOP job each command is executed as soon as it is input from the terminal. This enables the user to assess the effect of a command before deciding what command to issue next.

Scheduling

The GEORGE scheduling system schedules jobs on the basis of information supplied by the user. In the standard scheduling system this information consists of the urgency of the job, the earliest time at which it may be

started, and what peripherals the job requires. It is, however, possible for installations to modify the scheduling routines to take into account other factors more appropriate to their work load.

Accounting and budgeting

Each user is allocated budgets for the amount of the computer's resources available to him. GEORGE keeps a record of each user's consumption of resources and checks that the user does not exceed his budgets. The manager of a GEORGE installation can allocate his installation's resources by selling them to users for real or notional money. In addition, users can themselves sell part or all of their allocation to other users. Thus the user can be encouraged to make good use of his allocation.

At the end of a job GEORGE will output a summary of how much of his budgets the user has consumed, and at the end of an accounting period (specified by the installation manager) a full account is provided for each user.

THE OPERATOR'S FUNCTION

The operator working in an installation that is controlled by GEORGE will find that his function is different from that in an installation run under Executive. The purpose of this manual is to describe those activities which the operator will have to carry out in order that the GEORGE system may function efficiently.

The following list gives the main duties of an operator in a GEORGE 3 environment:

- 1 Loading the system (see Chapter 2, page 7)
- 2 Communicating with the system (see *Operator Communication with GEORGE and Executive*, below).
- 3 Helping the system to process the workload (see Chapter 2.1)
- 4 Handling peripherals (see Chapter 3, page 23). In particular:
 - (a) Making peripherals available to the system
 - (b) Loading input documents on to available basic peripherals
 - (c) Unloading output documents from basic peripherals
 - (d) Loading magnetic tapes and freeing magnetic tape decks
 - (e) Loading direct access devices
 - (f) Assisting repeats on card and paper tape readers
- 5 Controlling the number of pool and work tapes available to the system by means of the NEW and DEAD commands (see Chapter 8, pages 90 and 118).
- 6 Controlling the dumping system (see Chapter 4, page 43). In particular:
 - (a) Processing the dump tapes at regular intervals
 - (b) Providing the system with a suitable increment for a general restore in the event of a serious breakdown
- 7 Communicating with user's jobs, in particular:
 - (a) Answering questions output by users' jobs (see the ANSWER command in Chapter 8)
 - (b) Receiving informative output from users' jobs
- 8 BROADCASTing of messages to users where appropriate.
- 9 Dealing with breakdowns and restarting the system (see Chapter 6).
- 10 Closing down the system (see Chapter 7).

OPERATOR COMMUNICATION WITH GEORGE AND EXECUTIVE

In an installation that is being run under GEORGE the operator will communicate with both Executive and GEORGE. Normally the operator will receive messages and requests from GEORGE and will issue commands to GEORGE. Any communication with Executive or the program being run will then be handled by GEORGE. However during the loading of GEORGE and in the case of a breakdown of GEORGE the operator will communicate directly with Executive.

Operator Communication with Executive

COMMUNICATION DURING LOADING OF GEORGE

Before GEORGE is loaded, but after GEORGE has been restored (see Chapter 2, page 7), the appropriate Executive should be loaded in the normal manner. Once loaded, Executive will identify itself by typing out a message on the operator's console. Executive will then request any modifications (except on 1903A, 1902S or 1903S processors) and the date and time in the usual way. If this information is successfully input by the operator, Executive will carry out the initialization procedures for the loading of GEORGE, and will then inform the operator that it is ready to commence the loading of GEORGE. The operator must type a message to Executive indicating the version of GEORGE that is to be loaded and the peripheral that is to be used for loading. The communication between Executive and the operator during the loading of GEORGE is described in detail in Chapter 2, page 7.

COMMUNICATION DURING GEORGE BREAKDOWNS

If the operator wishes to close down GEORGE, there are three ways he can do it and these are described in Chapter 7, page 63. The reloading of GEORGE after the operator has closed down the system or after a system breakdown has occurred is discussed fully in Chapter 6, page 59. Whatever the reason for reloading, the basic procedure will be the same.

The operator will communicate with Executive until GEORGE has been successfully reloaded. In the case of a system breakdown Executive will first inform the operator what type of breakdown has occurred, and will also cause GEORGE to print out a postmortem, if required.

Operator communication with GEORGE

Once an installation is running under GEORGE the operator will receive two kinds of system output on his console, informative output and system requests. In addition, certain of the GEORGE commands can be used by the operator.

INFORMATIVE OUTPUT

The informative messages received by the operator from GEORGE are described in this manual in the accounts of the procedures during which they are likely to occur. For reference, the informative messages are also listed in alphabetical order in Appendix 1 together with their meanings and the action to be taken if necessary.

SYSTEM REQUESTS

When a system request is output the job that has generated it is halted awaiting operator action. The request is repeated at regular intervals (determined by the installation parameter MESSAGE TIME, see the INST PARA command, Chapter 8) and the job remains halted until the operator takes some action. It is therefore important that the operator should take action on system requests as quickly as possible. There are two, or in some cases, three courses of action the operator can take when he receives a system request:

- 1 He can implement the request; this will cause the job to continue from the point it has reached
- 2 He can issue a CANTDO command; this will cause GEORGE to take the default action that is appropriate to the activity that generated the request
- 3 In the case of requests concerned with basic peripherals only, he can issue a TERMINATE command; this will terminate the activity that generated the request. In some cases this command has the same effect as CANTDO

(In certain rare circumstances a system request may be repeated after the operator has taken action on it. In this case the operator should ignore the repeated request.)

System requests are described in the manual in the accounts of the procedures during which they are likely to occur. For reference they are also listed in alphabetical order in Appendix 1, page 157, together with their meanings and the alternative actions that can be taken.

OPERATOR COMMANDS

The GEORGE commands that the operator can issue from his console are described in Chapter 8, page 65. The commands that an operator can issue from the nominated console of a cluster (other than the central cluster) are listed under *Commands available to the remote operator*, page 42.4.

CANCELLING CHARACTERS ON THE CONSOLE

It is possible for the operator to cancel characters typed incorrectly while he is communicating with GEORGE. If a ← character is typed, the previous character will be cancelled; the ← character may be repeated n times to cancel the previous n characters of a line. Note that this facility is available only while the operator is communicating with GEORGE, not while communicating with Executive.

GEORGE ERROR MESSAGES

System error messages will be output to the operator's console, but these are of interest only to ICL staff analysing postmortems. Some of these messages are GEORGE error messages, which are distinguished by the prefix GEOERR.

Chapter 2 Setting up GEORGE

CONTENTS OF THE TAPE ICLKG3MARK n (OR ICLKG4MARK n)

The installation is supplied with a magnetic tape called ICLKG3MARK n (for GEORGE 3) or ICLKG4MARK n (for GEORGE 4). The contents of this tape are as follows:

- 1 A binary dump of a magnetic tape Allocator/Loader program.
- 2 GIN.
- 3 A binary version of GEORGE.

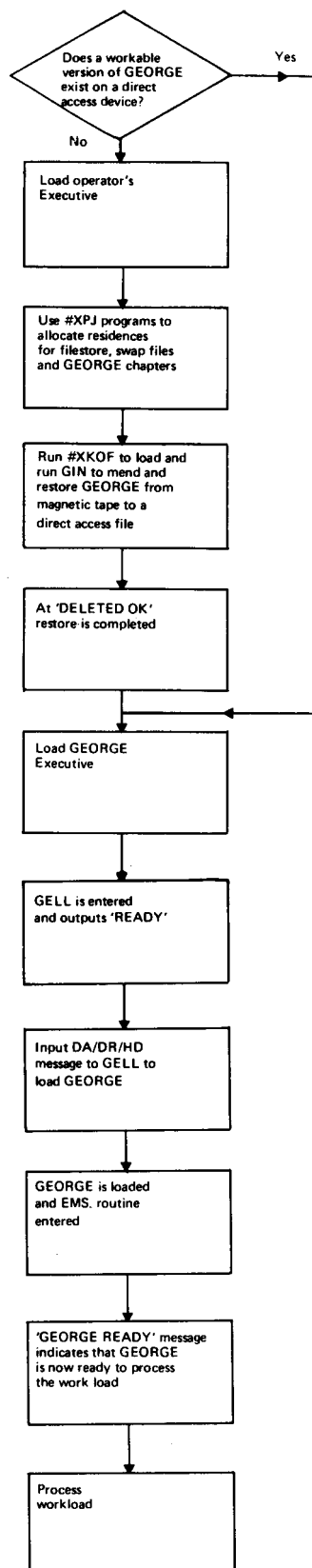
WHAT SETTING UP GEORGE INVOLVES

In order to set up a working version of GEORGE the following processes need to be considered:

- 1 Because GEORGE can be loaded only from a direct access device the binary version of GEORGE held on the ICLKG3MARK n (or ICLKG4MARK n) tape must first be *restored* to a previously created direct access backing store file.
- 2 If any *mends* (amendments) have been issued for the mark of GEORGE being set up, they must be incorporated in GEORGE using the GIN program.
- 3 The following special exofiles need to be allocated before GEORGE is loaded (see 4, below).
 - (a) A residence, known as the *GEORGE chapter residence*, to hold GEORGE.
 - (b) One or more residences, known as *swap residences* to contain object programs that have been temporarily removed from core store to make room for other programs to be run.
 - (c) One or more residences, known as *filestore residences* to contain the GEORGE filestore.
- 4 Once the required version of GEORGE is held on a direct access file and the various residences have been allocated, GEORGE can be loaded and the Early Morning Start (E.M.S.) routine set in motion.

The method and order of implementation of the above processes differs according to whether:

- 1 The installation manager requires to allocate the various residences, restore and incorporate any mends to GEORGE using an operator's Executive, prior to loading a GEORGE Executive and then loading GEORGE (see *Setting up GEORGE with an operator's Executive*, page 9).
- 2 The whole procedure for setting up GEORGE is to be carried out using a GEORGE Executive (see *Setting up GEORGE with a GEORGE Executive*, page 13).



Flowchart of setting up GEORGE with an operator's Executive

SETTING UP GEORGE WITH AN OPERATOR'S EXECUTIVE

If GEORGE is held on a U.D.A.S. device, or a drum that has not been cleared since the last GEORGE session and is known to contain the required version of GEORGE, and there are no mends to be incorporated, there is no need to load the operator's Executive at all. The procedure to be followed commences at *Loading the GEORGE Executive*, page 11.

If a new issue of GEORGE is being set up from magnetic tape or some mends need to be incorporated, the procedure to be followed is described below.

The procedure for setting up GEORGE with an operator's Executive is summarized in the flowchart opposite.

Restoring GEORGE to a direct access device

The operator's Executive should be loaded in the usual way and the #XPJ programs used to allocate residences for the filestore, swap files and the GEORGE chapters (see Chapter 2, of the manual *GEORGE 3 and 4 Operation Management*).

The GIN program is loaded into core store by means of the bootstrap loader program, #XKOF, which is supplied as a binary dump on cards or paper tape (it should be noted that #XKOF ignores the magnetic tape Allocator/Loader program); GIN then restores GEORGE.

In order to restore GEORGE to direct access backing store and incorporate the mends, the installation manager therefore needs to make up a pack of cards or a paper tape consisting of the following:

- 1 The loader program #XKOF.
- 2 Steering lines for #XKOF.
- 3 Certain GIN directives giving details of the direct access file that is used to hold GEORGE.
- 4 GIN directives to include any amendments to GEORGE.

THE LOADER PROGRAM #XKOF

The operator loads #XKOF by pressing the F button on the console typewriter associated with the device on which the restore pack/tape has been loaded. On a 1902S, 1903A or 1903S central processor the restore pack has to be loaded by the operator typing LOAD and GO (see the manual *1901/2/3 Operating (with console typewriter)* Edition 3, TP 4287).

Steering lines for #XKOF

The steering lines for #XKOF must include an OFR directive and a RES directive (see Chapter 3 of the manual *GEORGE 3 and 4 Operation Management*).

Error messages

One of the following messages will be output if #XKOF halts because of an error:

Message	Meaning	Action
HALTED:- CP, CR, LP, TP or TR	A peripheral is required (perhaps because one of the specified type has not been engaged)	The operator should make one of the specified type available, by engaging it, for example, and then type GO #XKOF
HALTED:- NOT ENOUGH CORE AVAILABLE	More core store is required	The operator should make more available and then type GO #XKOF
ILLEGAL FILENAME ON STEERING LINE	The file name or subfile name does not begin with a letter	The operator should correct the error, if possible, and re-engage the reader.
WRONG DIRECTIVE ON STEERING LINE		The operator should correct the steering line, if possible, and re-engage the reader.
subfile SUBFILE FORMAT IS INCORRECT	These messages cause the tape to be closed and the program to be deleted	When the cause of the error has been detected the operator should re-submit the card pack/tape
S/F subfilename IS NOT ON filename		
TAPE FAIL ON MTn		

THE GIN PROGRAM

When the GIN program has been loaded it will read certain control directives which are specified and described in Appendix 1 of the manual *GEORGE 3 and 4 Operation Management*.

When GIN finishes, it deletes itself and the message

DELETED OK

is output to the operator's console. The process of restoring GEORGE is then completed. The time taken by this process depends on the number of mends being incorporated but will usually be at least 10 minutes.

Operator messages

During a restore of GEORGE many operator messages are theoretically possible. The few which are likely to occur are given below. In general, any other message indicates an unusual situation such as corrupted tapes or a hardware failure. In such cases, the run should be started again, from the beginning. It is always advisable to double check the cards in the restore pack in such cases.

The following messages will occur during a normal run:

<i>Message</i>	<i>Meaning</i>	<i>Action</i>
DISPLAY RP	The user program file has been successfully restored.	None. This message always appears in a normal run.
DISPLAY RW	The user workfile has been successfully restored.	None. This message always appears in a normal run.
DELETED OK	Successful end of run.	Go on to next job.

The following messages, which may occur during a run, indicate exception conditions:

<i>Message</i>	<i>Meaning</i>	<i>Action</i>
DELETED CC	#COR card has improper format.	Correct card and rerun job.
HALTED CL	GIN cannot get the core it requires.	Make more core available, and GO.
DELETED FI	The last directive listed is in error and compilation cannot continue.	Correct card and rerun job.
DISPLAY LP UNIT X FIX	GIN has disengaged the printer because 'paper low' indicator has been set.	Deal with the paper low situation and re-allocate printer. Program will continue automatically.
HALTED PC	Disc file not available.	Either make required cartridge available and GO, or check card for mis-punching of filename, correct the card and re-run the job.
HALTED PL	Not enough drum to contain GEORGE file.	Clear drum and GO, or abandon job and reload Executive clearing drum.
HALTED PO	Not enough drum or disc to contain GEORGE file.	Re-run with a larger value on the #PRO card.
HALTED PT	Disc or drum: There is a temporary hitch connected with the number of files open, or other programs having system files open, or integrity code failure.	Can happen either just after loading Executive or the disc cartridge on which the file is, or when multiprogramming. GO after a minute or two, or after other programs using same backing store have finished.
DELETED WA	No scratch space at all was available on the appropriate device type.	Clear drum or load scratch disc as appropriate

<i>Message</i>	<i>Meaning</i>	<i>Action</i>
HALTED WI	Workfile invalid. The device type given in the #WOR directive is not acceptable.	Correct card and re-run job.
HALTED WT	Either too many files are open, or system files are open on the workfile device.	As for HALTED PT.
DELETED WU	Workfile used up.	Increase the size of workfile given in the #WOR directive or, if workfile is on disc, check that a sufficiently large scratch area is available. If it is not, load a cartridge with a larger scratch area. Re-run job.

Loading the GEORGE Executive

Once the required version of GEORGE is held in a direct access file, the loading process can begin. The GEORGE Executive is loaded in the usual way, and once loaded, it will output an identifying message on the console typewriter. If the Executive contains the appropriate facility, this identification will be followed by a request for modifications (except on 1903A, 1902S or 1903S processors), and, having received these, Executive will request the date and time. These must be supplied by the operator in the usual format. If the operator uses an incorrect format or an invalid combination, Executive will retype the request.

When the GEORGE Executive has been successfully initialized, it will set up and enter a routine called GELL (GEorge Loader Loader), (see below).

Loading GEORGE

GELL

After GELL has been entered, READY is output on the operator's console to indicate that the operator should specify the version of GEORGE that is to be loaded. He does this by typing the appropriate one of the following messages:

- 1 HD *filename* for a 2851 drum
 - 2 DR *filename* for a 1964 drum
 - 3 DA *filename*
- or
- DA *storage unit serial number, filename* for a U.D.A.S. device

It should be noted that the following formats are still acceptable for U.D.A.S. devices:

- (a) FD *filename*
- (b) ED *filename*

or

ED *storage unit serial number, filename*

HD/DR/DA/ED/FD may be extended by any number of alphabetic characters, for example, DRUM instead of DR.

storage unit serial number is of the form *nnnnnn

filename is the name of the file containing GEORGE and may be up to twelve characters in length although it must commence with an alphabetic character.

There must be at least one space between HD/DR/DA/ED/FD (plus extension, if any) and *filename* or *storage unit serial number*.

Examples

DRUM GEORGE-3.MK7

DA GFILEONE

After the message has been read, one of the following replies will be output:

<i>Message</i>	<i>Meaning</i>	<i>Action</i>
OK	The message has been accepted.	
ERROR B	The message contains incorrect parameters	The operator must try to retype the message correctly.
ERROR C	There is no such message	
ERROR J	The requested peripheral is not available.	

When the message has been accepted, the file indexes of all devices of the type indicated by the code will be searched for the named file. When the file is found, the first part of it will be loaded into core; this is the GEORGE loader. If the program is successfully loaded, the following message will be output:

LOADED

If the program is not successfully loaded, one of the following messages will be output by GELL:

<i>Message</i>	<i>Meaning</i>	<i>Action</i>
FAULT M	The named file cannot be found in the file index of any device of the given type	The operator should attempt to find the device containing the file, load it and retype the HD/DR/DA/ED/FD message
UNIT <i>nn</i> :-FAIL	A parity error has been found in flaw index on disc or there is no entry in index for failed block	Try loading from a different file
UNIT <i>nn</i> :-FAULT N	System control area format is wrong	The operator can attempt recovery by retyping the HD/DR/DA/ED/FD message but this time referring to another appropriate file
UNIT <i>nn</i> :-FAULT P	Auxiliary control area format is wrong	The operator can attempt recovery by retyping the HD/DR/DA/ED/FD message but this time referring to another appropriate file

Executive will output FIX messages for any devices of the given type that are inoperable and not WRONGed if it cannot find the file on the operable ones.

This message takes the form

UNIT *nn*:-FIX

The operator should engage the relevant device.

THE GEORGE LOADER

When the GEORGE loader has been successfully loaded it proceeds to load the fixed core segments of GEORGE from the backing store file containing GEORGE and to initialize the core store. During this process one of the following messages may be output:

<i>Message</i>	<i>Meaning</i>	<i>Action</i>
BAD COMPILATION	The backing store address of GEORGE's fixed core does not correspond to the type of device to which GEORGE has been restored. This will occur if, for example, a drum compilation has been restored to disc	The operator should obtain either a correct compilation or else restore to the correct device
GEORGE BSTYPE UNKNOWN	Loader has been given an unknown device type for the GEORGE device	The operator should take an Executive postmortem and attempt to reload with a new Executive
GEORGE FIXED CORE TOO LARGE FOR THIS EXEC (GEORGE 4 only)	Self-explanatory. Probably the wrong combination of Executive and GEORGE has been loaded	The operator should ensure that the correct versions of GEORGE and Executive are used together

<i>Message</i>	<i>Meaning</i>	<i>Action</i>
TRANSFER FAILURE ON LOADING; PLEASE RELOAD	Whilst loading the fixed core segments, a transfer failure has occurred	GEORGE will then break (without giving a postmortem) with the message SYSTEM DELETED. The GEORGE Executive must be reloaded and a further attempt made to reload the same or a different version of GEORGE.
UNIT <i>nn</i> FAIL	A transfer failure has occurred whilst reading from unit <i>nn</i> (backing store devices only)	None. This message will be followed by the message TRANSFER FAILURE ON LOADING; PLEASE RELOAD
UNIT <i>nn</i> FIX	Whilst loading the fixed core segments from unit <i>nn</i> , the latter has gone inoperable	Operator should engage the device. The loader will loop until he does
VERSIONS OF GEORGE AND EXEC INCOMPATIBLE		GEORGE breaks (without giving a postmortem). Operator must reload correct GEORGE Executive and load a compatible version of GEORGE

When the loader has successfully loaded GEORGE's fixed core segments and initialized the core store, the Early Morning Start routine will be entered (see *Early Morning Start routine (E.M.S.)*, page 22).

SETTING UP GEORGE WITH A GEORGE EXECUTIVE

If GEORGE is held on a U.D.A.S. device or a drum that has not been cleared since the last GEORGE session and is known to contain the required version of GEORGE the procedure to be followed to set up a working version of GEORGE is described under *Loading GEORGE*, page 11.

If a new issue of GEORGE is being set up from magnetic tape the procedure to be followed is described under *Allocating the filestore residences, swap residences and GEORGE chapter residence using the Allocator/Loader*, below, followed by the section *Restoring and loading GEORGE using the Allocator/Loader*, page 16.

If there are any mends, they must be incorporated in GEORGE using the GIN program. Since the GIN program will not run under a GEORGE Executive, if the installation has no operator's Executive, GIN must be run under a working GEORGE whenever mends have to be incorporated to a version of GEORGE held on magnetic tape (possibly the same as the working version). The procedure to be followed is described under *Mending GEORGE*, page 21.

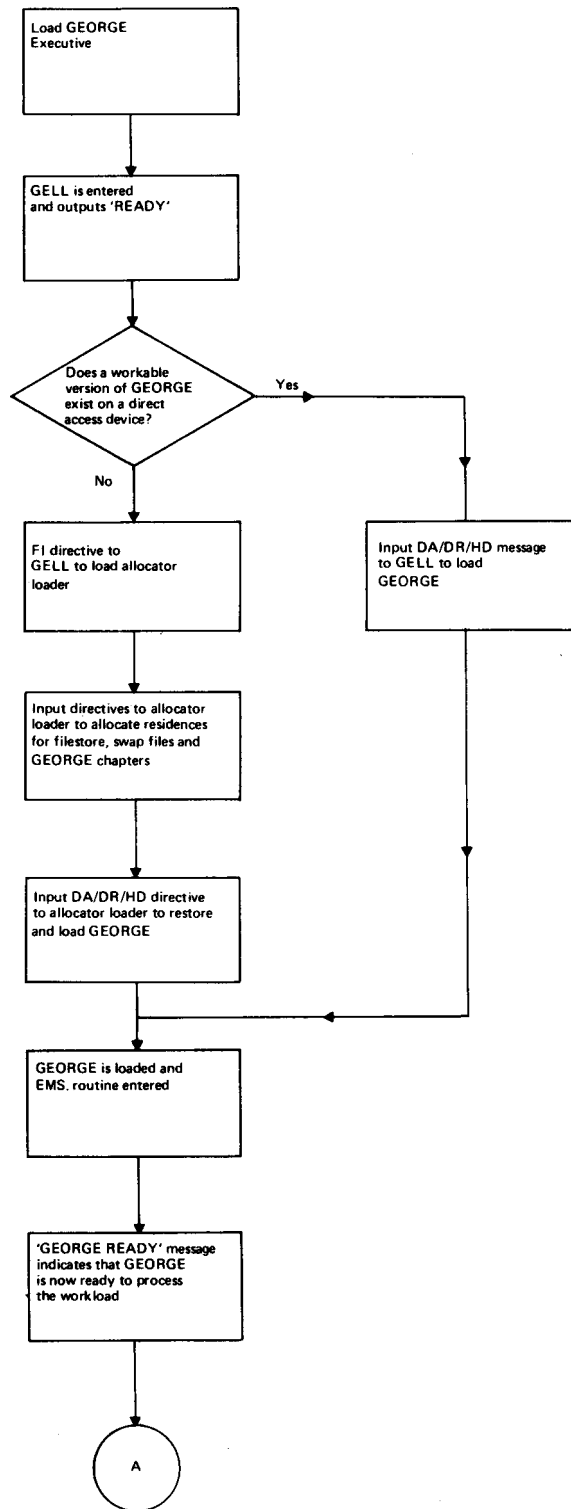
The procedure for setting up GEORGE with a GEORGE Executive is summarized in the flowchart on pages 14 and 15.

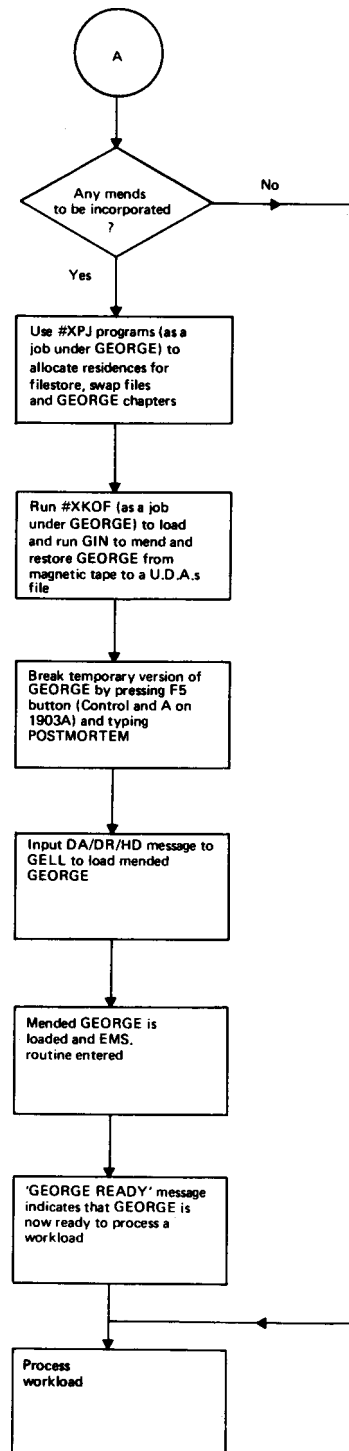
Allocating the filestore residences, swap residences and the GEORGE chapter residences using the Allocator/Loader

The process of allocating any residences (filestore, swap or GEORGE chapter) that do not already exist, and restoring and loading GEORGE is carried out by the special magnetic tape Allocator/Loader program which is held in a separate subfile at the front of the ICLKG3MARK n (for GEORGE 3) or ICLKG4MARK n (for GEORGE 4) tape. Once the Allocator/Loader is loaded into core all input to it by means of directives; full descriptions of the latter are given in Chapter 3 of the manual *GEORGE 3 and 4 Operation Management*.

The sequence for allocating residences is as follows:

- 1 The GEORGE Executive is loaded in the usual way, and once loaded, it will output an identifying message on the console typewriter. If the Executive contains the appropriate facility, this identification will be followed by a request for modification (except on 1903A, 1902S or 1903S processors), and, having received these, Executive will request the date and time. These must be supplied by the operator in the usual format. If the operator uses an incorrect format or an invalid combination, Executive will retype the request. When the GEORGE Executive has been successfully initialized it will set up and enter a routine called GELL (GEORGE Loader Loader). After GELL has been entered, READY is output on the operator's console.
- 2 The tape containing the Allocator/Loader and GEORGE is loaded on deck *m*. This tape will normally be called ICLKG3MARK n (or ICLKG4MARK n) but may be renamed if necessary.





Setting up GEORGE with a GEORGE Executive

- 3 If a 2851 drum is required, the operator should ensure that it is connected. He should also ensure that all backing store units required for filestore residences, swap residences and the GEORGE chapter residences are on-line.

- 4 When GELL outputs the message READY, the operator should issue the following message to load the magnetic tape Allocator/Loader:

FI *m name*

where *name* is the name of the subfile containing the required version of GEORGE and may be up to 12 characters long

m is the number of the deck on which the tape is loaded

- 5 The Allocator/Loader is loaded into core and the message

LOADED

is output followed by two new lines.

- 6 Directives can now be input to allow the Allocator/Loader program to allocate the exofiles required for the chapter residence, filestore residences and swap residences. These directives will be supplied by the installation manager and their formats and functions are fully described in Chapter 3 of the manual *GEORGE 3 and 4 Operation Management*. The directives may be issued either from the operator's console, or from a card reader or paper tape reader. If directives are to be issued from a basic peripheral, it is first necessary to issue a CR or TR directive from the operator's console. These directives have the formats

CR *n*

or

TR *n*

where *n* is the unit number of the card or paper tape reader being used.

All subsequent directives after a CR or TR directive are read from the specified peripheral. After each directive is read it is output to the operator's console, followed by OK if it is correct and can be executed successfully. If it is incorrect or cannot be executed, an error message will be output after the directive (see *Allocator/Loader directive error messages*, page 17). If this happens and if input is from a card reader or paper tape reader, the reader will be disengaged. The directive can then be corrected by either typing it correctly on the operator's console and repeating the TR or CR directive (to continue reading further directives from the slow reader) or by correcting the relevant card or paper tape and then resubmitting it by retyping the TR or CR directive.

Restoring and loading GEORGE using the Allocator/Loader

When the filestore residences, swap residences and GEORGE chapter residence have been allocated, one of the following directives, input via the operator's console to the Allocator/Loader, will cause GEORGE first to be copied to the backing store file from which it is to run and then to be loaded:

- 1 HD *filename* for a 2851 drum
- 2 DR *filename* for a 1964 drum
- 3 DA *filename*

or

DA *storage unit serial number, filename* for a U.D.A.S. device

It should be noted that the following formats are still acceptable for U.D.A.S. devices:

- (a) FD *filename*
- (b) ED *filename*

or

ED *storage unit serial number, filename*

HD/DR/ED/FD/DA may be extended by any number of alphabetic characters, for example, DRUM instead of DR. *storage unit serial number* is of the form *nnnnnn and *file name* is an on-line file large enough to hold GEORGE.

There must be at least one space between HD/DR/DA/ED/FD and *file name* or *storage unit serial number*.

When the reply ^{OK}
LOADED has been given to the HD/DR/DA directive, GEORGE has been transferred from magnetic tape to the specified backing store file and control is passed to the usual loading mechanism as described under *The GEORGE loader*, page 12. This is followed automatically by the Early Morning Start routine (see page 22) at the end of which the message GEORGE READY is output followed by the time and date.

A working version of GEORGE has now been set up. If any mends are to be incorporated into GEORGE this version of GEORGE may be used as a temporary working version under which the re-allocation for swap file and filestore residences and the mending of GEORGE can be carried out as jobs. This is described under *Mending GEORGE*, page 21.

Allocator/Loader directive error messages

The following is a list of error messages which may be output to the operator's console after a directive has been submitted to the Allocator/Loader, together with their meanings and the action required.

<i>Message</i>	<i>Meaning</i>	<i>Action</i>
BAD COMPILATION	Whilst loading fixed core from disc, Loader has an invalid value for the backing store address of fixed core.	This could occur if a drum compilation was being restored and then loaded from disc. It will be followed by the PLEASE RELOAD message.
CARTRIDGE/UNIT NOT FOUND	Output after CT/DA directive; specified storage unit not found on installation.	Load and/or engage required disc and resubmit directive; or check that correct storage unit is named in directive and, if not, resubmit corrected directive.
COMMAND/FORMAT ERROR	A directive with incorrect format or invalid parameter(s) has been submitted.	If the directive was input from the console, it should be retyped correctly. If from a card or paper-tape reader, that reader will be disengaged. The Allocator/Loader will now accept directives only from the operator's console, so that amongst the possibilities open are: (a) to continue reading directives from the card or paper tape reader by re-engaging the reader and retyping the CR/TR directive (b) correcting the invalid directive by typing the correct directive on the console and continuing as in (a).
DISC NOT NEWLY INITIALISED	Output after CT directive. The Allocator/Loader will only allocate files on newly initialised storage units (that is, no files have been allocated on the disc since it was last initialised).	Try again with a newly initialised storage unit.
DURING RELOCATION OF FILES	This message will follow a UNIT <i>nn</i> FAIL message, where <i>nn</i> is the operator's number of the drum on which relocation of files has failed during execution of a DE directive.	The files on the drum system concerned should be assumed corrupt.
FILENAME ALREADY EXISTS	Output after FI or DS directive. The filename specified already appears in a drum directory or	Use a different name, delete the existing one or just carry on.

<i>Message</i>	<i>Meaning</i>	<i>Action</i>
	has been requested in a preceding DS directive, respectively.	
FILE NOT FOUND	1 Output after DE directive because file to be deleted cannot be found 2 Output after DA/HD/DR directives because the backing store file to which GEORGE is to be transferred from MT cannot be found.	The user should create the necessary file and try submitting the DA/HD/DR directive again.
FILE REQUESTED TOO BIG	Output after FI directive. The file of the specified size could not be accommodated on any drum of the specified type (regardless of free space)	Check directive input.
GEORGE BSTYPE UNKNOWN	Loader has been given an unknown device type for the GEORGE device	The operator should take an Executive postmortem and attempt to reload with a new Executive
INVALID CYL.NO(S)	Output after DS directive because: 1 Starting cylinder requested for the first file is not the first cylinder after system file 2 Starting cylinder number greater than ending cylinder number. 3 Starting cylinder number is not one greater than the ending cylinder number of the previous DS directive. 4 A cylinder number out of range for the storage unit concerned has been specified	Resubmit corrected DS directive, or start again from scratch with CT directive.
INVALID FILENAME	Output after a DS/FS/HD/DR/DA/DE/FI directive. The specified filename (a) consists of more than 12 characters (b) does not commence with a letter (c) is not present when it should be.	Correct the directive and resubmit.
INVALID SERIAL NO	After a CT/FS/DA directive. One of the following errors has been detected in the specified serial number: (a) it consists of more than 6 digits (b) it contains a non-octal digit	Correct the directive and resubmit.

<i>Message</i>	<i>Meaning</i>	<i>Action</i>
	(c) it is not present when it should be.	
LOST MY TAPE	Output after a DA/HD/DR directive. The tape from which the Allocator/Loader came has become disengaged before transcription of GEORGE	Reload the GEORGE Executive and try again, perhaps with tape on another unit.
MT CHECKSUM FAIL	Output after a DA/HD/DR command. A block read from the tape has a checksum fail.	This will be followed by a PLEASE RELOAD message and then GEORGE breaks (without entering postmortem). (See <i>PLEASE RELOAD</i> message.)
MT FAIL	Output after DA/DR/HD directive. Whilst reading GEORGE from magnetic tape a failure has occurred.	This will be followed by a PLEASE RELOAD message and then GEORGE breaks (without entering postmortem). (See <i>PLEASE RELOAD</i> message.)
NO ONLINE { HSD 1964 DRUM	No on-line drum of the type indicated has been found.	Engage drum(s) required and resubmit the directive.
NO ROOM ON HSD	Output after FI directive for the 2851 drum. The Allocator/Loader has searched the 2851 drum systems without finding one with sufficient free space to accommodate the file requested.	Delete sufficient files on the 2851 drum or clear it to make room, or make available an additional 2851 drum, if possible.
NO ROOM ON 1964 DRUM	Output after FI directive for the 1964 drum. The Allocator/Loader has searched the 1964 drum systems without finding one with sufficient free space to accommodate the file requested.	Delete sufficient files on the 1964 drum or clear it to make room.
NO SWAP RESIDENCE DECLARED	Output to DA/HD/DR directive	Submit an FS directive for the swap residence(s) and resubmit DA/HD/DR directive.
NOT UDAS DISC	Output after CT directive. The program checks that the disc is to U.D.A.S. standards before checking whether it is initialized.	Check that correct serial number was specified. If not, retype directive. Otherwise resubmit directive naming a U.D.A.S. device.
PLEASE RELOAD	This message is output after an unrecoverable error has occurred. Such errors will be first reported by their respective messages.	GEORGE breaks (without entering postmortem). The GEORGE Executive must be reloaded and the procedure for setting up GEORGE restarted.
TRANSFER FAILURE ON LOADING; PLEASE RELOAD	The Loader is loading the fixed core segments and a transfer failure has occurred.	GEORGE will then break with the message SYSTEM DELETED. The GEORGE Executive must be reloaded and a further attempt made to reload the same or a different version of GEORGE.
REPOSITIONING OF TAPE FAILED	The attempt to reposition the tape after a backing store failure after a DA/DR/HD directive has failed. It follows a UNIT <i>nn</i> FAIL message	This will be followed by a PLEASE RELOAD message and then GEORGE breaks (without entering Postmortem) (See <i>PLEASE RELOAD</i> message)
RESIDENCE 3 NOT DECLARED	Output after DA/HD/DR directive	Submit an FS directive for residence 3 and resubmit DA/HD/DR directive.

<i>Message</i>	<i>Meaning</i>	<i>Action</i>
TAPE REPOSITIONED	The attempt to reposition the tape after a backing store failure after a DA/DR/HD directive was successful. It will follow a UNIT <i>nn</i> FAIL message	The DA/HD/DR directive may be resubmitted (perhaps to a new backing store file).
UNIT <i>nn</i> FAIL	(a) A failure has occurred on a card or paper tape reader (b) A failure has occurred on a transfer on a backing store device. The action of the Allocator/Loader in this situation is governed by the importance of the transfer. If the transfer failure has occurred during the search for a file or storage unit, then the message is purely informative and the Allocator/Loader continues its search. If the transfer failure occurs during the transcription of GEORGE to its backing store file, an attempt is made to reposition the tape so that the loading directive may be resubmitted or an alternative file used for GEORGE. If the transfer failure occurs during the rewriting of drum directories or during relocation of drum files, a further informative message is output and execution of the directive terminated. If the failure occurs during transfers involving system control areas (file indexes) after a CT or EF directive, execution of the directive is terminated and, in the case of the EF directive, the system control area should be assumed corrupted. If the message is output just after Loader has been loaded (indicated by the OK reply to the DA/DR/HD directive and the message LOADED), a transfer failure has occurred whilst Loader is loading fixed core. It will be followed by the message TRANSFER FAILURE ON LOADING; PLEASE RELOAD.	This will be followed by a UNIT <i>nn</i> FIX message. See immediately below.
UNIT <i>nn</i> FIX	(a) A card or paper tape reader being used by the Allocator/Loader has been disengaged (b) A backing store device being used by the Allocator/Loader has been disengaged. This may occur if, for example, the device holding the file to which GEORGE is being restored after a DA/DR/HD directive is put off-line	The Allocator/Loader will loop until the specified reader is re-engaged. The device should be re-engaged. The Allocator/Loader will wait until this has been done before continuing with execution of the directive.
UNITS <i>nn mm</i> ... INOPERABLE	Output after DA/CT directive if the specified storage unit cannot be found or after DA <i>filename</i> if the specified file cannot be found. This message gives a list of inoperable disc transports for information. It is followed by the message giving the cause of the error (CARTRIDGE/UNIT NOT FOUND or FILE NOT FOUND)	Informative message

<i>Message</i>	<i>Meaning</i>	<i>Action</i>
WHILST REWRITING DIRECTORY	This message will follow a UNIT <i>nn</i> FAIL message, where <i>nn</i> is the operator's number of the drum whose directory is being rewritten during execution of an FI or DE directive.	The directory of the drum system concerned should be assumed corrupt.
WRONG CONTEXT	Output after DS or EF directive because: 1 A DS directive has not been preceded by a valid CT directive 2 An EF directive has not been preceded by a valid CT directive and at least one valid DS directive.	
VERSIONS OF GEORGE AND EXEC INCOMPATIBLE		GEORGE breaks (without entering postmortem). The operator must reload the correct GEORGE Executive and load a compatible version of GEORGE.

Mending GEORGE

When a temporary working version of GEORGE has been set up and the GEORGE READY message output, the #XPJ programs can be run under this version of GEORGE to re-allocate, as necessary, residences for the filestore, swap file and GEORGE chapters of the mended version of GEORGE being produced. The GIN program is loaded into core by means of the bootstrap loader program #XKOF (it should be noted that #XKOF ignores the magnetic tape Allocator/Loader). #XKOF should be held in a filestore file and the restore and mend procedure run as a job under the temporary working version of GEORGE as follows:

- 1 Load #XKOF.
- 2 Submit steering lines for #XKOF.
- 3 Submit GIN directives giving details of the direct access file that is to be used to hold the mended version of GEORGE.
- 4 Submit GIN directives to include any amendments to GEORGE.

THE LOADER PROGRAM #XKOF

The steering lines for #XKOF must include an OFR directive and a RES directive (see Chapter 3 of the manual *GEORGE 3 and 4 Operation Management*). One of the following messages will be output if #XKOF halts because of an error.

<i>Message</i>	<i>Meaning</i>	<i>Action</i>
HALTED:-CP, CR, LP, TP or TR	A peripheral is required (perhaps because one of the specified type has not been engaged).	The operator should make one of the specified type available, by engaging it for example, and then type GO.
HALTED:- NOT ENOUGH CORE AVAILABLE	More core store is required.	The operator should make more available and then type GO
ILLEGAL FILENAME ON	The file name or subfile name does not begin with a letter.	The operator should correct the error if possible and re-engage the reader.

<i>Message</i>	<i>Meaning</i>	<i>Action</i>
subfile SUBFILE FORMAT IS INCORRECT	These messages cause the tape to be closed and the program to be deleted.	When the cause of the error has been detected the operator should re-submit the card pack/tape.
S/F <i>subfile name</i> IS NOT ON <i>filename</i>		
TAPE FAIL ON MTn		

THE GIN PROGRAM

When the GIN program has been loaded it will read certain control directives which are specified and described in Appendix 1 of the manual *GEORGE 3 and 4 Operation Management*. A list of error messages that may be output on the operator's console during the running of the GIN program is given on page 10.

When GIN finishes, it deletes itself and the message

DELETED OK

is output to the monitoring file. The process of restoring the amended version of GEORGE is then completed. The time taken by this process depends on the number of mends being incorporated but will usually be at least 10 minutes.

At this point two workable versions of GEORGE exist; the temporary working version and the newly mended version in a direct access backing store file. To transfer control to the newly mended version of GEORGE the operator should first press the F5 button (or Control and the appropriate other key on a 1903A, 1902S or 1903S) on his console and then type POSTMORTEM (or just POST) in response to Executive's console message "EXEC MESSAGE:". A postmortem will usually be unnecessary as the working GEORGE is being broken only in order to transfer control to the new mended version.

When the normal postmortem procedure is finished (see Chapter 5, page 57) GELL is re-entered and outputs the following message on the operator's console:

GELL READY

The operator should now load the newly mended version of GEORGE by following the procedure described under *Loading GEORGE*, page 11. This is followed automatically by the Early Morning Start routine (see below) at the end of which the message GEORGE READY is output which will be followed by the date, time and current value of GEORGE Mean Time. (GEORGE Mean Time (GMT) is the time, in minutes, that the machine has been running since 1st January, 1972).

EARLY MORNING START ROUTINE (E.M.S.)

When GELL has successfully loaded the file containing GEORGE, the Early Morning Start (E.M.S.) routine will be entered. This routine, amongst its other functions, will search for GEORGE's filestore residences. These files will be on one or more direct access devices, and will have been specified in advance during the restore process. Within GEORGE, there is a residence file table which contains an entry for each residence stating its residence number, file name, the type of device that the file is held on and, when the file is held on a U.D.A.S. device, its storage unit serial number if this was originally specified.

All devices containing GEORGE's filestore residences should be on-line at loading time. If GEORGE cannot locate a filestore residence, the operator will be notified by the console message:

FILENAME *filename* NOT FOUND

IS IT AVAILABLE?

where *filename* is the name of the file that cannot be located.

If the requested residence is not available, or if it has been decided not to include it in the filestore, the operator should type NO (or N). GEORGE will then continue without the missing file, unless it is residence number 3 without which GEORGE cannot continue. If, however, the file is available but the device containing it has not been engaged or loaded, the operator should engage the relevant device and reply YES (or Y). GEORGE will then search the device for the missing residence and if it still cannot be found, the operator will again be notified and asked about its availability.

If in reply to GEORGE's question the operator replies YES and then realizes that he meant NO he should

immediately type NO. If this NO is typed before the engage button is pressed, GEORGE will continue as if the operator has originally typed NO.

If GEORGE's residence file table contains an entry for a file giving the storage unit serial number of the cartridge containing that file, and the cartridge cannot be located, then the operator will receive the message:

CARTRIDGE *nnnnnn NOT FOUND

IS IT AVAILABLE?

where *nnnnnn is the storage unit serial number. The operator procedure is the same as in the case of a missing file (above).

If there is a transfer failure or if the system control areas of a specified cartridge or the first fixed disc is corrupted, then the operator will be informed by the message:

FAILURES ON UNIT *nnnnnn THEREFORE

FILENAME filename NOT FOUND

IS IT AVAILABLE?

If the fault cannot be rectified it will probably mean that one or more of the residences will not be available.

If any swap residence is found not to be a multiple of the sheetsize, GEORGE will reduce the size of that residence in order that it is and will inform the operator with the following message:

RESIDENCE m REDUCED BY nK WORDS

Once GEORGE has investigated all declared filestore residences, it will output the message:

RESTORE?

The operator may do one of the following:

1 (a) Type Y (or YES).

This indicates that a general restore is necessary using the most recent uncorrupted increment; that is, GEORGE is to restore the filestore from the last suitable dump. (The procedures for dumping and restoring are described in Chapter 6, page 59).

The following message is then output on the console:

INCREMENT?

and the subsequent action is as described under *General restore routine*, page 44.

(b) Input the number of the latest increment directly.

When GEORGE is loaded for the first time the restore will be made from the dump tape supplied with the GEORGE tape. This dump contains an initial filestore and has an increment number of 1.

GEORGE will then terminate its Early Morning Start routine by outputting the message:

GEORGE READY

which will be followed by the date, time and current value of GEORGE Mean Time. (GEORGE Mean Time (GMT) is the time, in minutes, that the machine has been running since 1st January, 1972). This informs the operator that GEORGE is now ready to start processing the workload.

2 Type N (or NO).

GEORGE will examine the filestore residences and decide whether they contain a usable filestore. If they do, GEORGE will terminate its Early Morning Start routine by outputting the operator message:

GEORGE READY

which will be followed by the date, time and current value of GEORGE Mean Time. (GEORGE Mean Time (GMT) is the time, in minutes, that the machine has been running since 1st January, 1972). This message informs the operator that GEORGE is now ready to start processing the work load.

If the contents of the filestore residences are unrecognizable, the action is as described under *Hardware, other software, operator or system limitation errors*, page 62.1.

During E.M.S. the file :SYSTEM.PROPERTY is investigated to make sure that it is a Mark 7 version. If it is not, a

Mark 7 version with only the property CENTRAL in it is created (see *The property system*, Chapter 11 of the manual *GEORGE 3 and 4 Operation Management*) and the following message output on the central operator's console:

:SYSTEM.PROPERTY RECONSTRUCTED—ONLY PROPERTY 'CENTRAL' NOW EXISTS

A comparison between :SYSTEM.PROPERTY and :SYSTEM.IDF is then carried out to check that the two files are compatible with respect to properties attributed to remote devices and the existence of console properties (see *Console property* page 30). If any discrepancy is discovered, :SYSTEM.IDF is adjusted to match

:SYSTEM.PROPERTY and the following message output on the central operator's console after the GEORGE READY message:

**:SYSTEM.IDF HAS BEEN ADJUSTED TO MAKE IT COMPATIBLE WITH :SYSTEM.PROPERTY
(AN IDFLIST IS ADVISED)**

An IDFLIST is advised so that the exact contents of :SYSTEM.IDF are available for reference.

If the installation has any message buffering multiplexers connected to it (see Chapter 3, page 40) GEORGE will ask if the Installation Description File, if it exists, needs updating for each message buffering multiplexer by outputting the question:

IS A CONFIGURATION UPDATE REQUIRED ON UNIT *n*?

to the operator's console. The operator may reply YES or NO. If a configuration update is required, GEORGE will ensure that the unit is MOPped OFF and then outputs the message:

UNIT *n* MOP OFF COMPLETED

if the unit was previously in the MOP ON state. GEORGE will then obtain a configuration report from the DCP and use it to update the I.D.F. (see page 42). This action will be completed before the GEORGE READY message is output, that is, before processing can commence. The operator should ensure that the Inter Processor Buffer is operable and that the correct DCP is running in the communications processor before loading GEORGE).

At no time prior to the GEORGE READY message should any peripheral engage buttons be pushed unless the operator is specifically requested to do so because they may be ignored and no guarantee can be given about when or whether they will be acted upon.

Chapter 2.1 Processing the workload

JOBS

Once GEORGE has been set up and started the operator's function is to help GEORGE to process the workload as efficiently as possible. The workload consists of a number of jobs. Jobs are of two kinds, user-issued jobs and system-issued jobs.

User-issued jobs may be divided into the following three categories:

- 1 A job defined by a job description that is input to the system on cards or paper tape, headed by a JOB command and terminated (normally) by four asterisks. Jobs of this kind are normally run as soon as the operator has loaded the job description on to a peripheral of the required type and has engaged the peripheral (however, see *Job descriptions*, page 24).
- 2 A job defined by a job description that is input to the system on cards or paper tape, headed by an INPUT command and terminated (normally) by four asterisks. When the operator engages the peripheral on which the job description is loaded, the job description is read and stored within the system, but the job is not normally run until a RUNJOB command is issued. RUNJOB commands can be issued from the operator's console, from card or paper tape readers, or from within other jobs. It is possible to use one job description of this kind to run a number of jobs, each initiated by a different RUNJOB command.
- 3 A MOP job, that is, a job run from a MOP console. Jobs of this kind concern the operator only in so far as program input/output operations in a MOP job may involve the use of peripherals.

The WHATJOBS and WHATSTATE commands

The operator can find out what jobs are currently in the system and obtain information about their progress by means of the WHATJOBS and WHATSTATE commands. The format of these commands, and categories of information that they provide, are given in Chapter 8.

It should be noted that until the output from a WHATJOBS, WHATSTATE, WHATLIST, WHATPER or WHATPROPS command is complete any subsequent such command will be held waiting. (WHATLIST, WHATPER and WHATPROPS are used to obtain information about peripherals.) An excessively long queue of such waiting commands will slow the system down significantly.

THE OPERATOR'S FUNCTION

It is GEORGE's function to process each job from start to finish with as little human intervention as possible. The operator is mainly concerned with introducing jobs, inputting data, loading tapes and discs and handling output; in other words, servicing peripherals. These tasks are described in Chapter 3. However, there are occasions when the operator is required to assist in the management of the workload in other ways. These are described in this chapter.

INSTALLATION PARAMETERS

GEORGE is likely to be required to work differently in different installations and with different types of workload. The installation manager can indicate his requirements to GEORGE by specifying the value of certain system parameters called *installation parameters*. A complete list of these is given in Chapter 8 in the specification of the INSTPARA command, which is the command used to change their value.

Certain installation parameters are referred to elsewhere in this manual when they affect an activity which is being described; for example DUMPTIME and DUMPTAPES affect dumping, and are referred to in Chapter 4. Operator action required in connection with installation parameters which concern the running of jobs is described in this chapter.

Control of on-line backing store

DEFAULTBS

The amount of on-line backing store allowed to a job (see *Operating Systems GEORGE 3 and 4*, Chapter 3) is determined by the installation parameter DEFAULTBS. If a job attempts to exceed this limit the message

LIMIT OF ONLINE BACKING STORE REACHED

will be output to the central operator's console, together with the username and jobname of the job concerned. This will be followed by the message

CAN THIS JOB'S LIMIT BE INCREASED?

Operator action depends on whether the job specified is one that is permitted by the installation manager to exceed the current value of DEFAULTBS, as follows:

If the job is permitted to exceed the current value of DEFAULTBS, the operator should take the following steps, in the order given:

- 1 Increase the value of DEFAULTBS by means of the INSTPARA command to the value permitted by the installation manager for this job

- 2 Type YES (or Y) on the operator's console. The message

LIMIT INCREASED TO n KWORDS

will then be output.

- 3 After this message is output, restore DEFAULTBS to its normal value.

Notes

- 1 It is essential that DEFAULTBS is changed before YES is typed; otherwise the results are likely to be the same as if NO had been typed (see below).
- 2 No new job should be introduced to the system between steps (a) and (c) above, since a job is allowed the amount of backing store specified by DEFAULTBS at the time that the job is introduced.

If the job is not permitted to exceed the current value of DEFAULTBS, the operator may either

- 1 Type NO (or N) on the operator's console. Appropriate action will then be taken by the job.

or

- 2 ABANDON the job.

Siting of object programs in core (GEORGE 3 only)

Failure by the system to swap an object program into core may result in unjamming action, as described in Chapter 7 of the manual *GEORGE 3 and 4 Operation Management*. This is influenced by the installation parameters CHAPTERQUOTA, OBJECTQUOTA, BACKGROUND and MOPLIMIT. One of the following messages will be output:

- 1 PROGRAM OF JOB *jobnumber* TOO BIG FOR SWAPIN UNDER PRESENT SYSTEM LOAD

Operator action should be as determined by the installation manager (see *GEORGE 3 and 4 Operation Management*, Chapter 7). If the program is subsequently swapped in, either as a result of operator action or because of a change in the system load, the message

PROGRAM OF JOB *jobnumber* SUCCESSFULLY SWAPPED IN

will be output.

- 2 (a) SYSTEM OVERLOAD: UNABLE TO SATISFY OBJECTQUOTA BY n

where n is a number of words.

- (b) REALTIME PROGRAM(S) OF JOB(S) *jobnumber*₁, *jobnumber*₂, . . . *jobnumber* _{n} PREVENTING SWAPIN BY FRAGMENTING CORE

Operator action in either of the above cases should be as determined by the installation manager. If the quota can subsequently be met, either as a result of operator action or because of a change in the system load, the message:

SYSTEM CAN NOW MEET OBJECTQUOTA
will be output.

Chapter 3 Peripheral handling

INTRODUCTION

An installation running under GEORGE may have the following types of peripheral:

- 1 Magnetic media, such as tape decks, storage units and drums, that may be used by GEORGE or ONLINED to a user program.
- 2 Local basic peripherals which may be used by GEORGE or ONLINED to a user program.
- 3 Communications peripherals such as multiplexers or teleprocessors which may be used by GEORGE or ONLINED to a user program.
- 4 Terminal peripherals attached to the central computer via communications equipment
- 5 Other peripherals, termed *exotics*, that are never used by GEORGE but which may be used ONLINE to a user program.

While GEORGE is running, the main function of the operator is to handle peripherals so as to maintain satisfactory throughput. In particular, his duties are:

- 1 Loading input documents on to available basic peripherals.
- 2 Unloading output documents from basic peripherals.
- 3 Loading magnetic tapes and freeing magnetic tape decks.
- 4 Assisting repeats on card and paper tape readers.

Jobs

To perform his function efficiently the operator needs to understand what kinds of job can be run under GEORGE, and, in particular, the ways in which jobs can utilize peripherals. The three main types of job are:

- 1 A job defined by a job description that is input to the system on cards or paper tape, headed by a JOB command and terminated (normally) by four asterisks. Jobs of this kind are normally run as soon as the operator has loaded the job description on to a peripheral of the required type and has engaged the peripheral (however, see *Job descriptions*, page 24).
- 2 A job defined by a job description that is input to the system on cards or paper tape, headed by an INPUT command and terminated (normally) by four asterisks. When the operator engages the peripheral on which the job description is loaded, the job description is read and stored within the system, but the job is not normally run until a RUNJOB command is issued. RUNJOB commands can be issued from the operator's console, from card or paper tape readers, or from within other jobs. It is possible to use one job description of this kind to run a number of jobs, each initiated by a different RUNJOB command.
- 3 A MOP job, that is, a job run from a MOP console. Jobs of this kind concern the operator only in so far as program input/output operations in a MOP job may involve the use of peripherals.

The two types of peripheral used most in a job are basic peripherals and magnetic tapes. This chapter describes how these peripherals should be handled by the operator in a GEORGE environment.

A job may also involve the use of direct access devices, but their handling under GEORGE is on the whole the same as in an Executive controlled environment.

On-line and off-line

Input and output peripherals can be used in one of two ways: on-line or off-line.

ON-LINE

An on-line peripheral is connected directly to the object program concerned. On-line input is read straight from an on-line peripheral as it is required by the program. In the case of on-line output, results are written directly to a peripheral by the program that generates them.

OFF-LINE

Off-line peripheral channels are connected to the filestore, instead of to the program. Off-line input data is read from the peripheral into the filestore, where it is stored until it is erased by the user. The program can obtain the data from the filestore when it is required. Off-line data must be input to the system before the job requiring it is initiated. The peripheral involved will only be in use for the period during which the data is being read and filed by the system.

Similarly, off-line output data is written to the filestore, instead of directly to a peripheral.

BASIC PERIPHERALS

Input

Input on basic peripherals consists of both job descriptions and data. The input may be loaded on any basic peripheral of the required type.

JOB DESCRIPTIONS

A job description on cards or paper tape, headed by a JOB command, is loaded on to a peripheral in the normal way. The job is normally run as soon as the peripheral is engaged. However, it is possible that the limit on the number of such jobs in the system may have been reached. This happens if the number of these background jobs reaches both of the following values.

- 1 The limit BACKGROUND
- 2 JOBLIMIT minus the larger of
 - (a) MOPLIMIT
 - (b) the total number of MOP jobs

(For further details see Chapter 7 of the manual *GEORGE 3 and 4 Operation Management*)

For the values of BACKGROUND JOBLIMIT and MOPLIMIT, see the INSTPARA command (Chapter 8).

If the limit on the number of these jobs is reached, then the reading of further cards or paper tape is suspended until another job ends. If the peripheral needs to be used for other operations, the operator can issue an UNLOAD command and remove the job description from the device. The operator can discover the state of the peripheral prior to doing this by making use of WHATPER command.

A job description introduced by INPUT is loaded in the same way as off-line data (see below).

OFF-LINE DATA

Off-line data is normally input on cards or paper tape headed by an INPUT command and terminated by four asterisks.

ON-LINE DATA

On-line data is input to the system on cards or paper tape, headed by a DOCUMENT command. Data introduced by DOCUMENT is known as a *document*.

The operator can load a document on to any peripheral of the required type. Once a document has been loaded on a peripheral and the peripheral engaged GEORGE reserves that peripheral for the document. The peripheral cannot thereafter be used for other input operations until it has been released by the program that is to read from it, or the operator issues an UNLOAD command (see Chapter 8).

A document should not normally be loaded until just before the job in question is to be run. However, it is not essential for a document to be loaded before it is required by a job. If a document is requested by a job before the operator has loaded it, the system request:

PLEASE LOAD DOCUMENT *document name*

is output. The operator then has two options:

- 1 He can load the specified document on to an available peripheral and engage the peripheral. The job will then continue.
- 2 He can issue a CANTDO command. This will generate an error, which will be dealt with by the job description.

Should the operator have an input document with no DOCUMENT command at the head of it, he may simulate a DOCUMENT command by typing a REDON command on his console, and then load the document.

A further possibility is that the operator may load a document, correctly headed, and the document may never be required by a job. Even if the job in question has been completed or abandoned or is never run, the peripheral will remain committed to the document loaded on it. In this case the operator should check, if necessary, that the job is complete and should then issue an UNLOAD command. This will free the peripheral on which the document is loaded. The document may then be removed and the peripheral used for other operations.

Note that if a system breakdown occurs while a job is running it may be necessary to reload both the job description and the program data in order to re-run the job. How much will have to be reloaded depends on how many of the files involved in the job will be restored when the system is reloaded. It is therefore recommended to keep the cards and paper tape necessary to run a job until the job has been completed and all the expected output has been produced.

FAILURES DURING INPUT

If a failure occurs during input of data or a job description, the following system request is output:

PLEASE ENGAGE UNIT *peripheral description* FOR ERROR RESTART

The operator has three options:

- 1 He can correct the failure if possible, and re-engage the peripheral. If the failure recurs, the messages will be repeated. Note that after an error restart on a card reader, the operator should always re-feed the last card to be read.
- 2 He can issue a CANTDO command. The effect of this command depends on whether the peripheral is being used on-line or off-line:
 - (a) On-line: the failure will be dealt with according to the job description.
 - (b) Off-line: the file which is being produced will be erased.
- 3 He can issue a TERMINATE command. This is most useful when a read failure has occurred on a card or paper tape reader. Again, the effect of the command depends on whether the peripheral is being used on-line or off-line:
 - (a) On-line: the failure will be dealt with according to the job description.
 - (b) Off-line: the file being produced will be closed and its current contents preserved.

Note that the operator may not be asked to engage the peripheral after being informed that there has been an input failure because the program has been informed of the failure and will take appropriate action.

OTHER PERIPHERAL EVENTS

There are three other occurrences that will disengage a peripheral and generate a system request:

- 1 When the program which is reading directly from the peripheral attempts to read beyond the last card or line of paper tape, and no provision for this has been made in the program.
- 2 When a job description or data being input to the filestore does not have the required terminator so that attempts are made to read beyond the last card or line of paper tape.
- 3 When the operator manually disengages the peripheral.

The operator has three standard options in all of these cases:

- 1 He can re-engage the peripheral, having reloaded it if necessary.
- 2 He can issue a CANTDO command.
- 3 He can issue a TERMINATE command.

The effects of these actions are as for peripheral failures (above).

UNLOADING BASIC INPUT PERIPHERALS

If the operator wishes to free a basic input peripheral which is loaded with

- 1 A document headed by a DOCUMENT command

- 2 A job headed by a JOB command and waiting because JOBLIMIT has been reached
- 3 A job or data headed by a JOB or INPUT command and waiting for one of the following:
 - (a) A file to be erased
 - (b) A backing store jam to be cleared
 - (c) A file to be retrieved
 - (d) A file to become available

he may do so by means of the UNLOAD command (see Chapter 8).

Output

Both off-line and on-line output are preceded by a heading identifying the user and the job to which they belong (see *Heading formats*, page 27). After being written to a file, off-line output will probably be output to a basic peripheral by a LISTFILE command issued by the job. It is possible that the output may not be produced until some time after the job has finished, owing to a shortage of basic peripherals. However, the heading will indicate where the output should be sent. This also means that if a LISTFILE command cannot be carried out (for example, if the file to be listed has been lost) GEORGE will be unable to inform the monitoring file (see below) of the job that has issued the command, if that job has already finished. If this occurs, one of the following explanatory messages will be output by the system on the operator's console:

LISTFILE OF *filename* IN *username.jobname* HAS BEEN LOST DUE TO PERMANENT EXCLUSIVE PROPERTIES

LISTFILE OF *filename* IN *username.jobname* HAS BEEN LOST DUE TO LOST FILE
username.filename

LISTFILE OF *filename* IN *username.jobname* HAS BEEN LOST AS DEVICE NO LONGER EXISTS
IN CLUSTER *property*

These messages warn the operator that he should no longer expect to find the specified listing when collecting together the output for the job concerned. The operator should however pass on the message(s) to the user.

MONITORING FILE

In addition to on-line or off-line output, every job also has a *monitoring file* to which information concerning the job is written in the course of the job. At the end of a job, its monitoring file is listed on a line printer. Like normal off-line output, the listing is produced by the LISTFILE command and is preceded by a heading identifying the user and the job to which it belongs (see *Heading formats*, page 27).

ENGAGING BASIC OUTPUT PERIPHERALS

In general the operator should keep all output peripherals other than card and paper tape punches permanently engaged. In the case of a card punch, he should engage the peripheral only when requested to do so. If GEORGE needs an output peripheral and none of the required type is available, the system requests:

PLEASE ENGAGE UNIT *peripheral description*

is output, where *peripheral description* is defined on page 66 and refers to a disengaged peripheral of the required type. The operator has three options:

- 1 He can engage the peripheral.
- 2 He can issue a CANTDO command. This will cause the peripheral concerned to be WRONGed and the message

UNIT *peripheral description* WRONGED

to be output on the operator's console. The activity that generated the request will then wait until a peripheral of the required type becomes available. The peripheral can be made available for use again by issuing a RIGHT command for it.

- 3 He can issue a TERMINATE command. If the peripheral is on-line, the effect will be as for CANTDO, except that the peripheral will not be WRONGed. Otherwise, the activity that requested the peripheral will be terminated.

HEADING FORMATS

LISTFILE outputs three lines of headings at the start of a listing. These are preceded by two pages of blank paper, 30 inches of paper tape run-out or two blank cards as appropriate. In the case of the line printer, a line of asterisks with upward pointing arrows at either end is also output, so that the print hammers may be checked. If output is sent to the monitoring file system, no heading is given.

The lines of heading are as follows:

1 **#LISTING OF *username.jobname* PRODUCED ON *date* AT *time***

This gives the date and time when the file was last written to. These are given in the standard formats for dates and times.

2 **#OUTPUT BY LISTFILE IN '*username.jobname*' ON *date* AT *time***

The user name and job name are those of the job in which the LISTFILE command was issued. The date and time indicate when the listing was started.

3 **DOCUMENT *file description***

This is the file description that was given as the first parameter of the LISTFILE command. If the total number of characters in this line exceeds the maximum allowed for the output peripheral in question characters are lost from the end of the line.

If the file is a multifile, described in Chapter 3 of the manual *Operating Systems GEORGE 3 and 4*, each element will be listed separately with its own headings and terminators. The elements will be listed in the order they are produced; however, there may be other listings between the listings of consecutive elements.

In the case of a paper tape punch, legible headings are produced in the form:

filename FROM JOB *username.jobname*

A legible heading is one which can be read by the operator: appropriate holes are punched to represent each letter. These legible headings are preceded and followed by 30 inches of runout.

FAILURES DURING OUTPUT

An attempt is made to distinguish between 'error condition' failures and 'exception condition' failures that may occur during the implementation of a LISTFILE command. In general, error conditions (such as transfer errors) result from hardware failures, while exception conditions result from a variety of events such as paper low on a line printer or the operator pressing the HOLD button.

Error conditions give rise to an automatic restart corresponding to the *restart option* (either ALL or PAGE) specified in the LISTFILE command (see Chapter 12 of the manual *Operating Systems GEORGE 3 and 4*). This restart action is described below.

Exception conditions will cause a PLEASE ENGAGE message to be output. (In the case of paper low this is preceded by a PLEASE RELOAD message as described in *Paperlow*, page 28). The operator can then choose whether or not he wishes to restart. If, for example, the HOLD button was pressed simply to enable him to examine a listing, he will very likely prefer the listing to continue without a restart. If no restart is deemed necessary, then the device can be engaged and the listing will continue. If a restart is desired then the operator should type

CANTDO ENGAGE

This will cause the restart action to be taken, but note that the listing will not necessarily continue on the same device. Continuation on the same device can be forced by re-engaging the device and making sure that GEORGE cannot find any other suitable device; the property system can help achieve this, but procedures must be carefully worked out to avoid undesirable side-effects.

Restart action is as follows:

- 1 If neither ALL nor PAGE was specified in the LISTFILE command, only the failing transfer is repeated with no repetition of the headings.
- 2 If ALL was specified the whole listing, including the headings, is repeated.
- 3 If PAGE (which applies to a line printer only) was specified, the action is as follows. If either the value of the installation parameter IPAGE is 1 or only one page has been listed, then the page being listed at the time of the failure is re-output. Otherwise the following system request is output on the operator's console:

PLEASE GIVE PAGES FOR ERROR RESTARTS ON UNIT *n*

The operator can then either reply with the **PAGES** command (see Chapter 8) or the **CANTDO** command. If he uses the **CANTDO** command, only the page being listed at the time of the failure is re-output. If he uses the **PAGES** command, however, the number of pages specified is output.

- 4 If restart is required during the output of the headings, the listing is always completely restarted. If restart is required during the output of the terminators, only these are repeated and only then if they can be output on the device used for the listing, that is, if the operator types **CANTDO**, the output of the terminators is abandoned and the listing is considered completed.

PAPERLOW

If data is being output on-line, a **PAPERLOW** condition on a line printer does not generate any output to the operator. However, if the user does not deal with the **PAPERLOW** condition in his program then the line printer will become inoperable and the result will be as described under *Failures during output*, page 27.

If the line printer is off-line, a **PAPERLOW** condition will cause the informative message

RELOAD *number*

to be output and the peripheral to be disengaged. The operator is advised to reload and re-engage the specified line printer on receiving this message.

TERMINATION OF LISTINGS

In general, when a **LISTFILE** is suspended and there is an outstanding **PLEASE ENGAGE** message for the device, the operator can abandon the listing by typing a **TERMINATE** command. If it is desired to abandon a listing currently in progress, the operator should disengage the device, wait for the **PLEASE ENGAGE** message and type **TERMINATE**.

In two particular cases, however, **TERMINATE** has a different effect:

- 1 If the listing is interrupted during output of the headings, **TERMINATE** has the same effect as **CANTDO**, that is, an attempt will be made to repeat the listing later.
- 2 During the listing of a **SETUP** file, **TERMINATE** is taken as implying that the paper is correctly set up, and the listing of the main file will be started as soon as the printer is re-engaged.

OUTPUT TO SPECIAL STATIONERY

When special stationery is used, a test pattern (specified by the job producing the listing) is repeatedly printed to allow the operator to align the paper correctly. When the paper is properly positioned the test pattern listing ends and the main output begins.

After each listing of the test pattern the printer disengages itself. The following message is output to the operator:

PLEASE LINE UP UNIT *peripheral description*

The operator may then line up the paper and engage the printer. The test pattern will again be printed and the printer will disengage itself. The operator may repeat this process until the paper is satisfactorily lined up. No further line up messages will be output unless an interval of **MESSAGETIME** (installation parameter) elapses before the printer is re-engaged. When the paper is aligned correctly the operator should issue a **TERMINATE** command for the peripheral. When the printer is re-engaged the main file will be listed.

If the operator is asked to **ATTRIBUTE** a special stationery property to a printer (see *Peripheral properties*, page 29), it will not be necessary to **WRONG** the printer as it is marked allocated before the request is output. However, a device should be **WRONGED** if the attributions of properties to the device are being changed and the system has not requested such changes. It will not usually be possible for the operator to anticipate the property requirements of a device but he may know, for example, that no other job requires the special stationery which has just been used and may be able to save time by changing the stationery immediately rather than waiting for the system to request him to do so.

UNLOADING BASIC OUTPUT PERIPHERALS

When a batch of output to a local card or paper tape punch is terminated, the message

UNLOAD *peripheral description*

is output and the punch is disengaged. The operator is advised to remove the cards or paper taper from the specified punch before he next engages it. Otherwise it will be difficult to separate this batch from any batch that is subsequently output on the same punch.

No message is sent to the operator when a batch of output is terminated on a line printer.

At the end of the listing the LISTFILE command again outputs a line of asterisks with upward pointing arrows at either end (in the case of the line printer), one paper throw, 30 inches of paper tape run-out or two blank cards, as appropriate. If the listing was to the card punch, the punch is disengaged and the operator is informed that the listing is complete. For a paper tape punch, five feet of paper run-out is considered sufficient to indicate to the operator the break between two documents, and the paper tape punch is disengaged in the same manner as a card punch.

If the operator wishes to bring unwanted printing to an end, he can do so by disengaging the line printer and then issuing the TERMINATE command from his console when the system request for the peripheral to be engaged is output (see *Termination of listings*, page 28). This method can also be used to stop output on card and paper tape punches.

Peripheral properties

It is possible to define a set of peripheral properties and attribute these to local and remote basic output peripherals. The information GEORGE requires about peripheral properties is provided by PROPERTY and ATTRIBUTE commands (see Chapter 8) issued from the operator's console. The PROPERTY command is used to define a property; this consists of a property name which may be attributed to any basic output peripheral. Thus a property GASBILLS may be declared with the intention of attributing this property to a line printer while it has special GASBILLS stationery loaded. A PROPERTY command is used to declare a property that exists throughout the installation and the actual connection of a declared property with a specific peripheral is done by issuing an ATTRIBUTE command with the required peripheral and property as its parameters.

PROPERTY NAME QUALIFIERS

The property name specified in the PROPERTY command may be given a pair of qualifiers as follows:

- 1 (a) **PERMANENT (PERM).** If this qualifier is given, any command that requests the property whilst it is not ATTRIBUTED will generate a command error. A PERMANENT property should be a property that will normally remain attributed to a device. For example, a property FAST, indicating a high speed line printer, would normally be a PERMANENT property.

or

- (b) **TEMPORARY (TEMP).** If neither PERMANENT nor TEMPORARY is given, TEMPORARY is taken as the default value. If TEMPORARY is given (explicitly or implicitly), any command that requests the property whilst it is not ATTRIBUTED to an available device of the required type, will cause the following system request to be output:

PLEASE ATTRIBUTE *property names* TO UNIT *peripheral description*

Having checked that the physical property is available the operator should then ATTRIBUTE the required property to the peripheral requested. A TEMPORARY property should be a property that will be frequently attributed to a device and shortly afterwards have this attribution cancelled. For example, a property PAPER, indicating special stationery, would normally be a TEMPORARY property that would be attributed only whilst special stationery was required

- 2 (a) **INCLUSIVE (INCL).** If this qualifier is given, devices to which the property has been ATTRIBUTED may also be used by commands that do not request the property

or

- (b) **EXCLUSIVE (EXCL).** If neither INCLUSIVE nor EXCLUSIVE is given, EXCLUSIVE is taken as the default value. If EXCLUSIVE is given (explicitly or implicitly), a unit to which the property is ATTRIBUTED may not be used by commands not requesting the property.

If a peripheral has TEMPORARY EXCLUSIVE properties not requested in a LISTFILE or ONLINE command then the operator will receive the following system request:

PLEASE FREE UNIT *peripheral description* OF PROPERTY *property name(s)*

When the operator receives this request he should issue the appropriate CANCEL ATTRIBUTE commands.

If a LISTFILE or ONLINE command also requires TEMPORARY properties which are not ATTRIBUTED to the device, the operator will receive the following system request:

PLEASE FREE UNIT *peripheral description* OF PROPERTY *property name(s)* AND ATTRIBUTE
PROPERTY *property name(s)*

The operator should then issue the appropriate CANCEL ATTRIBUTE command(s) followed by the appropriate ATTRIBUTE command(s).

If the operator types CANTDO to either of these requests (or to the request in 1(b), above) no further message about the LISTFILE command which caused the system request will be output for 30 minutes. However, should the operator subsequently comply with the system request before the message is repeated, the action will be recognized and the command causing the system request will be allowed to proceed. Thus, if the operator replies CANTDO to a PLEASE ATTRIBUTE message but later performs ATTRIBUTE on the peripheral, the LISTFILE command that needed a property ATTRIBUTED will be able to proceed. In the case of the ONLINE command error in ONLINE will be given if CANTDO is the reply to the system request.

Where possible all properties are remembered over a system break.

CONSOLE PROPERTIES

A property declared by means of a PROPERTY command in which the CONSOLE parameter is present is described as a console property. Console properties form the basis of peripheral clusters (see *Peripheral clusters*, page 42.2).

ATTRIBUTING PROPERTIES

A peripheral property may be attributed to a basic output peripheral by means of the ATTRIBUTE command (see Chapter 8). Several properties may be ATTRIBUTED to one peripheral, but to do this several ATTRIBUTE commands are required. Only one property may be ATTRIBUTED at a time, and the maximum number of attributions that may be made to any one device is five. An example of having two properties ATTRIBUTED to one peripheral might be when the PERM INCL property FAST and TEMP EXCL property GASBILLS indicating a high speed printer and special stationery, have both been declared by PROPERTY commands and are then both ATTRIBUTED to the same peripheral; this would ensure that any job wanting a high speed listing to this type of special stationery would use this printer. Console properties may not be ATTRIBUTED to local devices.

Note that a peripheral should always be WRONGed before issuing an ATTRIBUTE command for that peripheral unless the property attribution was requested by the system.

Although properties are remembered over a system break, only attributions of PERMANENT properties are reset at restart; attributions of TEMPORARY properties must be re-introduced.

CANCELLING PROPERTIES AND ATTRIBUTIONS

The attribution of a property to a peripheral is removed by the CANCEL ATTRIBUTE command (see Chapter 8).

The declaration of a property to the system is cancelled by a CANCEL PROPERTY command (see Chapter 8). When the property has been cancelled the following message is output on the operator's console:

PROPERTY *property name* HAS BEEN CANCELLED

If however GEORGE is still using information relating to a property (for example, if a LISTFILE request involving the property has not yet been performed) then any attempt to cancel the property will generate a command error.

EXAMPLES OF THE USE OF THE PROPERTY SYSTEM

Maintenance engineer

The problem of testing a peripheral under the control of GEORGE, that has just been serviced, is that unless the peripheral is in some way reserved for the engineer the peripheral may be used by GEORGE for some other purpose.

A simple method of allowing peripherals to be reserved for maintenance is to define a PERMANENT and EXCLUSIVE property (perhaps called MAINTENANCE or ENGINEER). When it is required to service and test a peripheral, the engineer should carry out the following procedure:

- 1 WRONG the required peripheral. This will ensure that GEORGE will not attempt to use this device

- 2 Service the device
- 3 ATTRIBUTE the EXCLUSIVE maintenance property
- 4 RIGHT the device to make it available again
- 5 ONLINE the device to the test program, requesting the EXCLUSIVE property in the ONLINE command

If a high speed or 1964 drum is ONLINED to a test program in this way, the following message will be output on the operator's console:

DRUM *n* IS ONLINE TO JOB *jobnumber*

If the object program that issued the ONLINE command writes to the drum, one of the following messages will be output:

DRUM *n* FREE

DRUM *n* FREE: PLEASE ALLOCATE FILES

DRUM *n* CORRUPT – PLEASE INITIALIZE

In the latter case, the drum should be initialized by using the Allocator/Loader CL directive (see Chapter 3 of the manual *GEORGE 3 and 4 Operation Management*).

Note that GEORGE provides no data security for ONLINED drums and therefore this facility should only be used for engineers' test programs.

Using special stationery on line printers

When a property name is to be used to allow special stationery to be loaded on a peripheral, the special stationery property should be EXCLUSIVE and TEMPORARY. The fact that it is TEMPORARY will allow GEORGE to ask the operator to ATTRIBUTE the property when it is required, and the operator should carry out the following procedures:

- 1 Load the special stationery
- 2 ATTRIBUTE the property

The job that requested the property may now proceed. (The procedure for dealing with the repeated listing of a test pattern to line up the stationery is dealt with under *Output to special stationery*, page 28). The property should be EXCLUSIVE or another job may be able to gain control of the printer before the operator has a chance to remove the special stationery after the current job has finished.

Alternatively, a PERMANENT and INCLUSIVE special stationery property may be declared. Such a property might be ATTRIBUTED to a printer that was temporarily loaded with 160 column paper so that jobs wanting this property could use this printer without preventing other jobs using it. Using a PERMANENT property means that the operator will not be asked to ATTRIBUTE this property to another printer if a job requests this property while the printer is being used by another job. Instead the second job will wait for the first job to release the printer.

Console properties and peripheral clusters

Console properties form the basis of peripheral clusters which are described more fully under *Peripheral clusters*, page 42.2. An example showing how a fairly complex communications system can be handled quite simply by using the cluster system can be found in the manual *GEORGE 3 and 4 Operation Management*.

MAGNETIC TAPES

Magnetic tapes are of two kinds, insecure and secure (librarian) tapes. A tape may be referred to by its name or serial number or both.

Insecure tapes

Insecure tapes are those tapes not known to the system. They may be subdivided into standard tapes (those having a standard 1900 Series header label) and non-standard tapes. The latter include virgin tapes, that is tapes that have not previously been used.

Standard insecure tapes may be introduced into the system by the operator by means of a NEW command, when they become secure.

If a standard insecure tape has a serial number that duplicates that of any other tape, it is classed as a *xenotape*.

The loading of xenotapes and non-standard tapes must be preceded by a REDON command.

An insecure tape may be given a header label by the LABEL command (although not normally if the tape is loaded on a 1971 deck).

Secure tapes

GEORGE keeps in the filestore a record of all secure tapes belonging to the installation and any insecure tapes currently loaded. The various files that contain these records and the routines for handling them are known collectively as the *librarian*. The central index of the librarian is a file called :SYSTEM.SERIAL which keeps a record of all secure tapes. Secure tapes are subdivided into three categories:

- 1 Owned tapes, that is tapes owned by specific users.
- 2 Pool tapes, that is tapes available for allocation to users.
- 3 Work tapes, that is tapes for temporary usage.

Apart from loading and unloading owned tapes when requested, the operator is concerned only with the handling of pool tapes and work tapes.

POOL TAPES AND WORK TAPES

Pool tapes and work tapes are both listed in :SYSTEM.SERIAL, where they are classed as 'pool tapes' or 'work tapes' respectively. The collection of all pool tapes known to the librarian is referred to as the *pool*, whereas work tapes are said to be held in the *worktape store*.

The operator can transfer insecure tapes to the pool or worktape store by means of the NEW command (see Chapter 8) and withdraw tapes from the pool or worktape store by means of the DEAD command. If a tape is to become a pool tape, the operator will be asked to load the tape (see *Loading magnetic tapes*, below) so that it may be relabelled POOL TAPE.

When a user introduces a tape into the system, the following message is output to the operator's console:

MT *tape serial number* IS NOW SECURE

When a user withdraws a tape from the system, the message

MT *tape serial number* IS NO LONGER SECURE

is output.

The console log therefore contains commands or messages for all tapes introduced to, or withdrawn from the system.

If the operator requires lists of tapes that have been input to the pool and the worktape stores he should use the LISTSERIAL and TAPES commands respectively. He can then fill a rack with suitable tapes and when all the tapes have been used the procedure can be repeated.

Loading magnetic tapes

When a magnetic tape that is not loaded is required, the following system request is output:

PLEASE LOAD $\left\{ \begin{array}{l} \text{MT } \textit{tape name} \\ \text{MT } \textit{tape serial number} \\ \text{POOL TAPE} \\ \text{WORK TAPE} \end{array} \right\} \textit{load condition}$

{ ON *deck characteristic* UNIT *number number* . . . :TSN *tsn₁ tsn₂*
 WHEN *deck characteristic* UNIT AVAILABLE

where POOL TAPE is a pool tape and WORK TAPE is a work tape; the *load condition* may be WPR (with write permit ring) or NO WPR (without write permit ring) and/or MOD*Number* (the mode of the deck that the tape is to be loaded on) or it may be omitted; the *deck characteristic* (if it is present) will be one of the following: TRACK7, TRACK9, PE or NRZI; the *numbers* are the operator's numbers of up to four available decks, and the *tsns* are two suggested serial numbers of tapes to load when the tape is requested by name only and more than one is available. (If the tape description is MT *tape name* then the *load condition* must be followed by FOR JOB *number*.)

The operator has three options:

- 1 He can load the tape as requested either on one of the decks suggested or on any other available deck. If a pool or work tape is requested, he can load any available pool or work tape.
- 2 He can issue a CANTDO command. The effect of the command depends on whether the tape was requested within a job description or within a program or by the incremental dumper:
 - (a) If the tape was requested within a job description then a command error is generated.
 - (b) If the tape was requested within a program then an error will be generated unless provision has been made for it to continue without the tape.
 - (c) If the tape was requested by the incremental dumper, the subsequent action is as described under *The incremental dumper*, page 47.
- 3 If for some reason the requested tape is unsuitable for use (for example, it may have been damaged), the operator may cause it to be marked as unavailable by issuing the TAPEWRONG command (see page 138). The tape remains unavailable for use and will not be requested again until a TAPERIGHT command (see page 136) is issued for the tape.

If the tape requested is a xenotape of a non-standard tape, a REDON command must be issued before the tape is loaded (see page 125).

If a user owns several tapes of the same name, ONLINEing a tape of that name will cause the serial numbers of two tapes to be suggested in the PLEASE LOAD request. The two tapes suggested will be two tapes in the user's directory which are not currently in use. However, if only one tape is not in use the serial number of a tape being used is also suggested. If all the tapes are in use the job will wait for one to become free. The operator need not load a suggested tape; any tape which fulfils the description and is in the user's directory is accepted.

If the operator loads a tape with the wrong load condition, one of the following system requests is output:

PLEASE LOAD WPR ON MT *tape serial number* ON UNIT *number*

PLEASE UNLOAD WPR ON MT *tape serial number* ON UNIT *number*

If the operator loads a pool tape or work tape without a WPR, then the tape is unloaded and the following informatory message is output:

UNIT *n* FREE: POOL/WORK TAPE NEEDS WPR

If the user has issued an ONLINE command for a non-standard tape or xenotape before that tape is loaded, the following message will be output:

PLEASE SELECT UNIT FOR $\left\{ \begin{array}{l} \text{NONSTANDARD} \\ \text{XENOTAPE} \end{array} \right\}$ *tape description* $\left\{ \begin{array}{l} \text{WPR} \\ \text{NO WPR} \end{array} \right\}$

MODEn FOR JOB NO *n*: *deck numbers*

The operator may not be asked to set MODE and WPR; up to four deck numbers are output. The operator should then issue the REDON command for the selected deck.

If a tape is loaded with a serial number duplicating one already loaded and a REDON command for a xenotape has not been issued, the following message will be output:

UNIT *n* FREE: MT *tsn* ALREADY ON UNIT *m*

If the operator loads the tape with the wrong WPR status, then one of the following system requests is output:

PLEASE $\left\{ \begin{array}{l} \text{LOAD} \\ \text{UNLOAD} \end{array} \right\}$ WPR ON MT ON UNIT *number*

If the operator successfully loads a tape, GEORGE will first attempt to read the header label. If GEORGE cannot read the header label, then one of two informative messages will be output. They both indicate that further operator action is necessary to implement the original system request.

- 1 UNIT *number* HEADER READ FAIL. RELOAD ON UNIT *number number* . . .

This message will be output if a failure occurs while the header label is being read and will be followed by the message

UNIT *number* FREE

The operator again has three options:

- (a) He can reload the tape as requested.
- (b) He can issue a CANTDO command. This will have the same effect as a CANTDO command after the original system request.
- (c) He can issue a TAPEWRONG command to make the tape unavailable for use.

2 NON STANDARD HEADER LABEL ON UNIT *n label*

where *label* is the first 24 characters that have been read from the header label. This message will be output if the tape does not have a standard 1900 Series header label and is therefore unacceptable to the system. It will be followed by the message

UNIT *number* FREE

If the operator's error had been to load the tape before issuing a REDON command, he should now issue the command and re-engage the deck: otherwise it will be necessary to issue a CANTDO command for the original system request. This will have the same effect as a CANTDO command issued immediately after that system request. The operator should remove the tape to free the deck. He may then either give this tape a label by using the LABEL command, and then reload it, or load another tape.

There is a third informative message that may be output during the reading of the header label:

LONG HEADER BLOCK ON UNIT *number*

This message means that the tape loaded on the deck with the specified operator's number has a header label that is more than 20 words long but is recognizable as a standard 1900 Series header label. The tape will be accepted by GEORGE but if it is renamed, words 20 onwards will be lost.

QUERY STATUS

When a secure tape that is also an owned tape is loaded, the name on the tape's header label is checked against the name, generation number and reel number in the directory entry. If they do not coincide, the tape is put into query status, until the situation has been remedied. A message is output on the operator's console indicating the state of :SYSTEM.SERIAL, the directory entry and the header label. The operator must inform the owner of the tape, so that he can take corrective action. The owner may either correct the directory entry by means of the RENAMEDIR command, or return the tape to the pool so that it may be NEWed by a new owner.

When a tape which is in query status is RETURNed to the pool, it retains its query status and cannot subsequently be picked up as a pool tape. It must first be claimed by a user issuing a NEW command naming the tape; this removes the tape from query status. He can then RETURN the tape if he has no use for it.

The query status message to the operator is only output at the time the tape is first put into query status. If an operator finds that the system is refusing to pick up a tape, he should carry out the following procedure:

- 1 Issue a WHATPER command to verify that the tape has been picked up by the system and to find out the name, reel number, and generation number of the tape, and, if it is owned, the owner of the tape.
- 2 If the tape has been picked up by the system, try to DEAD the tape. The resulting error message will tell if the tape is owned or in query status (that is, a RETURNed tape still in query status).
- 3
 - (a) If the tape is owned, the operator should communicate these details to the owner, who can then take appropriate action.
 - (b) If the tape is a RETURNed tape in query status, efforts should be made to establish the owner of the information on the tape, who can then claim the tape by issuing a NEW command naming the tape. If the owner of the information on a tape cannot be established, a NEW command naming the tape must be issued from within a job. The tape can then be RETURNed by the new owner and used as a pool tape.

MODE

If it is required to switch the mode of a deck, the operator may do so by means of the MODE command.

If the operator wishes to know which magnetic tape decks are available for the loading of tapes, he may issue the TRANSPORTS command, and a list of the available decks will be output.

Unloading magnetic tapes

A magnetic tape must not under any circumstances be unloaded unless GEORGE has output the message

UNIT *number* FREE

and then only if it is *certain* that the tape has not been re-engaged since the message.

The message may be output as a result of:

- 1 A tape failure
- 2 A TAPEWRONG command having been issued to the loaded tape
- 3 A job requesting that the tape be unloaded when it is released.
- 4 An UNLOAD command (see page 141)

In the first three cases, on receipt of the FREE message, the tape can and should be unloaded.

Case 4, which would be used when a deck is required for loading a new tape and no deck is known to be free, will generate a FREE message, or the message:

UNIT *number* IS NOT ENGAGED

or a message indicating that the tape is in use.

Provided that one of the first two messages is received, the tape may be unloaded. In the third case, another deck must be tried.

When GEORGE unloads a deck, the deck, unless it is a 1974, is disconnected from the processor. This means that, on decks other than 1974s, the deck address light will go out when the deck is free thus enabling the operator to differentiate between free decks and decks that are engaged but not in use. It also means that the operator need not partially unload a deck that has been freed by GEORGE before pressing the LOAD button.

Properties on magnetic tapes

Many installations have magnetic tape decks operating at different speeds and recording with different methods (7 track, 9 track phase encoded and 9 track NRZI). Usually when a job requests a tape it does not matter how the tape was recorded or how fast a deck it is loaded on, but there are cases where these factors could be important. For example, if a tape is to be used at another installation which only has 7 track decks, then it is obviously necessary for the job setting up the tape to be able to request a 7 track tape; or if a large file is to be sorted using worktapes to hold the intermediate sorted strings of records, it would be advantageous to be able to request fast decks for the tapes.

A number of special properties are used by the system to cater for the above situations. They are as follows:

<i>Property name</i>	<i>Meaning</i>
TRACK7	7 track tape (deck)
PE	9 track phase encoded tape (deck)
NRZI	9 track NRZI tape (deck)
TRACK9	9 track tape (deck)
MODE <i>n</i>	tape (deck) in recording mode <i>n</i>
NONS	nonstandard tape
FAST	fast deck
SLOW	slow deck

They can be specified as parameters to a command such as GETONLINE in the same way as property parameters can be included in LISTFILES for basic output peripherals. It is not necessary, however, to issue PROPERTY commands defining these property names, nor need they be ATTRIBUTED to appropriate decks; GEORGE is able to discern from information held internally which property names are relevant to which device. Note that the property names listed above are not reserved. The manager can issue PROPERTY commands defining them, ATTRIBUTE them to devices other than *MT and use them in the normal way. He can, for example use the property name FAST on a line printer.

The magnetic tape property system is therefore somewhat separate from other properties. The property names listed above can be used on devices other than *MT but with *MT only these names can be used and no others.

ENTRANT PROPERTIES AND PERIPHERAL PROPERTIES

A further difference between magnetic tape properties and basic peripheral properties is that the former can be subdivided into entrant properties and peripheral properties. An examination of the list of MT properties given above will reveal that many of them could refer either to a magnetic tape or to a magnetic tape deck. For example, TRACK7 could be either the property of a 7 track tape deck or the property of a tape that was last written to when loaded on a 7 track deck. The subdivision into *entrant properties* and *peripheral properties* reflects this dual role: entrant properties are the properties of tapes, peripheral properties are the properties of decks.

The entrant properties are: TRACK7,PE,NRZI,TRACK9,MODE n and NONS.

The peripheral properties are: TRACK7,PE,NRZI,TRACK9,MODE n ,FAST and SLOW.

REFERRING TO PROPERTIES ON MAGNETIC TAPES

The properties listed above may be referred to by the operator in commands such as NEW, TRANSPORTS and UNLOAD, either as a parameter or as a qualifier depending on the command.

Examples

NEW (36225,P) (*MT,PR TRACK7)	Introduces a 7 track tape to the pool.
TR PR FAST&MODE4	Requests a list of the fastest decks available that can be switched to mode 4.
UNLOAD *MT (PR TRACK7)	Frees all 7 track decks that are not in use.

UNUSABLE PERIPHERALS

If a basic peripheral or a magnetic tape deck becomes completely unusable then it must be removed from the system so that GEORGE will not attempt to use it until the fault has been corrected. The operator can do this by issuing the WRONG command (see page 156). Once this command has been issued GEORGE will no longer list that peripheral as part of the installation. When the peripheral is usable again, the operator can return it to GEORGE's control by means of the RIGHT command (see page 131).

DIRECT ACCESS DEVICES

Operator handling of direct access devices in a GEORGE environment differs from that in an operator's Executive controlled installation in the ways described below.

Loading direct access devices

When a direct access device that is not loaded is required, the operator receives the following system request:

PLEASE PUT STORAGE UNIT n ONLINE

The operator has two options:

- 1 He can load the specified device.
- 2 He can issue a CANTDO command. The job that requested the device will be informed that it is not available.

Unloading direct access devices

To unload a direct access device the HOLD button must be pressed. The panel light will be lit and GEORGE will check that the device is not in use. Provided that it is not in use the device will be disengaged. If it is in use the HOLD button will be ignored and the light will remain lit to indicate that an attempt has been made to disengage the device.

The STOP button must be used only in extreme emergencies as there is a danger that it may cause data to be lost.

Device failure messages

When a direct access device fails the following message is output:

UNIT *n* FAIL

where *n* is the geographical number of the device. This is followed by one of the messages below:

- 1 TRACK **n* BLOCK **m* (for an EDS, EDS30 or FDS)
- 2 WORD **n* (for a 1964 drum)
- 3 BLOCK **n* (for a high speed drum)

These are followed by a print of the 5-word control area involved in the failing PERI order (see below).

The 5-word control area print may be followed by the message:

PLEASE RE-ALLOCATE

This is output if a read transfer from a filestore residence fails or if position fail is reported on an FDS (see Bit 4 of Word 1 below). If the transfer was from an EDS or EDS30, GEORGE will disengage the device and the device need only be re-engaged for GEORGE to attempt the transfer again. For FDS and 1964 drums the unit will have to be put off-line and then on-line manually by the operator. GEORGE will then attempt the transfer again. If a read failure occurs on a high speed drum, the 5-word control area will be output and GEORGE will break with the message GEOERR TRANFAIL.

If a device becomes inoperable, the message:

UNIT *n* FIX

is output. The device should then be engaged and GEORGE will attempt another transfer. In this case no other information is output on the operator's console.

CONTROL AREA INVOLVED IN THE FAILING PERI ORDER

If UNIT *n* FAIL is output, the corresponding control area is as described below.

Word 0: Type/Mode

Bit 0=1	Always
Bits 1 to 8	Device type
Bits 9 to 23	Mode of transfer:
	=0 Read from device
	=1 Write to device

Word 1: Reply word

EDS and EDS30	Bit 0=1	Transfer in progress, other bits indeterminate (see notes)
	Bit 0=0	Transfer complete (see notes)
	Bit 1=1	Transfer terminated by end of cylinder (see notes)
	Bit 2=1	Track faulty
	Bit 3=1	Uncorrected error
	Bit 5=1	Inoperable
	Bits 6 to 23	Indeterminate (see note 2)
FDS	Bit 0=1	Transfer in progress: other bits indeterminate (see notes)
	Bit 0=0	Transfer complete (see notes)
	Bit 3=0 }	Transfer terminated by end of cylinder (see notes)
	Bit 1=1 }	

	Bit 3=0 } Bit 2=1 }	Address lost during data transfer
	Bit 1=0 } Bit 2=0 } Bit 3=1 }	Uncorrected error
	Bit 3=0 } Bit 4=1 }	Position fail on positioning
	Bit 5=1	Device inoperable or rejecting transfers
	Bits 6 to 23	Indeterminate (see note 2)
	Additional replies for the 2806 FDS are as follows:	
	Bit 1=1 } Bit 3=1 }	Standard interface parity fail (see note 1)
	Bit 2=1 } Bit 3=1 }	Multi-access failure: the device or channel cannot be accessed because it is allocated to another channel or processor (see notes)
1964 drum	Bit 0=1	Transfer in progress: other bits indeterminate (see notes)
	Bit 0=0	Transfer complete
	Bit 3=1	Error (repeat procedure exhausted)
	Bit 5=1	Inoperable
	Bits 6 to 23	Indeterminate
High speed drum (2851)	Bit 0=1	Transfer in progress: other bits indeterminate
	Bit 0=0	Transfer complete
	Bit 3=1 } Bit 1=0 } Bit 2=0 }	Uncorrected error
	Bit 1=1 } Bit 2=0 } Bit 3=1 }	Standard interface parity fail (see note 1)
	Bit 1=0 } Bit 2=1 } Bit 3=1 }	Multi-access failure: the device or channel cannot be accessed because it is allocated to another channel or processor (see notes)
	Bit 5=1	Device inoperable or rejecting transfers
	Bits 6 to 23	Indeterminate

Notes:

- The following replies will all result in GEOERR TRANFAIL after the control area has been printed:

Bit 0=1	Unterminated transfer
Bit 1=1	End of cylinder (but not always, see note 3 below)
Bit 2=1 } Bit 3=1 }	Multi-access failure (FDS and high speed drum)
Bit 1=1 } Bit 3=1 }	Standard interface parity fail

GEOERR TRANFAIL will also occur if a read transfer to a high speed drum fails.
- Bits 6 to 23 are meaningless unless Bit 1=1 in which case these bits contain the address of the word following the last word transferred.
- The end of cylinder error implies that the hardware has transferred an incorrect number of words. If it is

possible for GEORGE to recover from this situation the device is disengaged and then a request to reallocate the device is output (as for normal read failures).

Word 2

Amount of data to be transferred in words

Word 3

Address of the first word of the buffer area

Word 4

Device address. (The message above the 5-word control area is this word converted).

EDS	Bits 2 to 5	Most significant 4 bits of the head position
	Bits 8 to 11	Least significant 4 bits of the head position
	Bits 14 to 17	Head address
	Bits 21 to 23	Sector address
EDS30 and 2804 and 2805 FDS	Bits 0 to 11	Head position
	Bits 12 to 17	Surface
	Bits 18 to 23	Sector
2806 FDS	Bits 0 to 8	Head position
	Bits 9 to 14	Surface
	Bits 15 to 23	Sector
1964 drum	Bits 0 to 23	Starting word of the drum area involved in the transfer
High speed drum (2851)	Bits 0 to 23	Starting block of the drum area involved in the transfer

Exchanging filestore cartridges

Under normal running conditions it should not be necessary to move cartridges containing residences. If, however, a transport or spindle containing residences continually fails, the operator may move the cartridge from one transport or spindle to another, without breaking GEORGE, by first issuing a CARTRIDGE command (see *CARTRIDGE*, Chapter 8). The transports or spindles specified in the command will then be disengaged by GEORGE and the operator may unload the cartridge which contains the residence or residences and load it on the other transport or spindle specified in the command.

When the transport or spindle which should now contain the system cartridge has been re-engaged, GEORGE will read the storage unit's control header and check the serial number. If the correct cartridge has not been loaded on its new transport or spindle GEORGE will disengage the device and output the message:

UNIT *n* DISENGAGED – PLEASE LOAD CARTRIDGE *z* ON UNIT *n*

If the correct cartridge is loaded GEORGE will carry on reading and writing to the residence on its new transport or spindle.

COMMUNICATIONS SYSTEMS

Communications devices

GEORGE allows the use of two main types of communications devices, namely character buffering multiplexers and uniplexers and message buffering multiplexers.

CHARACTER BUFFERING DEVICES

The character buffering multiplexers that may be used with GEORGE are the 7007/2, 7920, 7930 character buffering devices.

Each *line* of a character buffering multiplexer is referred to by line number and these are used by GEORGE or by a program having such a multiplexer on-line.

A 7010/3 or 7010/5 uniplexer may also be used as a communications device with GEORGE.

MESSAGE BUFFERING DEVICES

The message buffering multiplexers used with GEORGE are processors of the 7900 Series. Either a 7901 or 7903 communications processor may be used. Each of these devices is a small computer running under the control of a Dedicated Control Program (DCP). The number of lines that can be attached to such a device is variable and depends on the nature of the devices attached to it and the amount of its core store available for the buffering of messages.

Each remote peripheral has a unique *identifier* input to the DCP when it is loaded. These identifiers are used by GEORGE or by a program having such a multiplexer on-line.

Terminal devices

There are three standard terminal devices used by GEORGE. These are 7020 remote data terminals, 1901A Remote Job Entry terminals, and 7071 teletypewriters. A 1901A RJE terminal is treated by GEORGE in almost exactly the same way as a 7020.

It is possible to connect other terminal devices to multiplexers running under GEORGE, but GEORGE does not deal with such devices as standard ones. Instead the user must **ONLINE** the multiplexer to a user written program that directly controls the non-standard terminal (see *Exotic peripherals*, page 42.5).

7020 REMOTE DATA TERMINALS

The 7020 data terminal may be used for Remote Job Entry (RJE) and for the transmission of bulk data. It consists of a data terminal to which a combination of the following peripherals may be attached:

- 1 Paper tape reader
- 2 Paper tape punch
- 3 Card reader
- 4 Line printer
- 5 7023 teletypewriter for use as an operator's console.

The bulk input devices are used to introduce job descriptions and user files into the filestore. Jobs may also be initiated from these devices by JOB or RUNJOB commands. The use of all basic peripherals attached to 7020 terminals is controlled by the *Peripheral cluster* system which is described on page 42.2.

1901A REMOTE JOB ENTRY TERMINALS

The peripherals that may be attached to a 1901A being used as a Remote Job Entry (RJE) terminal are the following:

- 1 Operator's console
- 2 Card reader
- 3 Line printer

As GEORGE treats a 1901A RJE terminal in almost exactly the same way as a 7020 all references in this chapter to 7020s may be taken to refer to 1901A RJE terminals as well.

TELETYPEWRITERS AND TERMIPRINTERS

The 7071 is the standard teletypewriter supplied by ICL for multiple on-line programming (MOP) work.

7083/1, 7084/1, 7085/1 and 7086/1 termiprinters can also be used as MOP terminals in a 7071 compatible mode (that is, with no lower case characters available).

System messages

- 1 If a system break occurs, indication is given to all MOP terminals and 7020 terminals.
- 2 When a multiplexer is MOPped OFF the message:

THIS TERMINAL HAS BEEN MOPPED OFF

is sent to all 7023 consoles.
- 3 When a 7020 line or identifier is ATTACHed to a conceptual multiplexer (see page 42.1) the message:

THIS TERMINAL HAS BEEN ATTACHED

is sent to the 7023 console concerned.

In cases 1 and 2 any 7023 input going on at the time the event happens will be cancelled and ignored.

Describing the communications configurations of an installation

GEORGE holds a list of local peripherals, each type identified by a number up to 63, in core. This list is a single level list with one entry per device. To enable GEORGE to hold a list of terminal devices attached to communications equipment, a list of more than one level is needed; the first level to describe the communications devices, the second level to describe the terminal devices attached. For efficiency this list is organized as a direct access file; thus it is not possible to write entries to this file using the built-in GEORGE editor. The file that holds information on communications equipment is known as the I.D.F. (Installations Description File).

The contents of the I.D.F. and its maintenance are described below, but before this is dealt with it is necessary to describe the manner in which the peripherals of an installation are uniquely identified.

PERIPHERAL IDENTIFICATION FROM OPERATOR'S CONSOLES

Local devices

As was stated above these are identified by a number up to 63, the conventional operator's unit number for the operator's console typewriter.

7020 devices with character buffering systems

When a 7020 terminal connected via a character buffering multiplexer is introduced to the system, it must be given a unique identifier. This identifier has the following format:

- 1 It must consist of two characters
- 2 The first character must be an alphabetic character
- 3 The second character must be alphanumeric

Every 7020 terminal has a set of interfaces which are numbered. One terminal peripheral is attached to each interface (a 7023 teletype requires two interfaces). Thus any peripheral attached to a 7020 may be uniquely identified throughout the installation by appending its 7020 interface number to its terminal identifier, giving a peripheral identifier of the following form:

IAA3

where the peripheral is attached to interface 3 of 7020 terminal AA. When GEORGE writes the 7020 information to the I.D.F., the unique identifier of a peripheral attached to a 7020 will be prefixed with the letter I. GEORGE will expect the operator to prefix any input message reference to a 7020 identifier with I.

GEORGE commands must be issued to identify peripherals attached to 7020 terminals.

7900 devices

Devices attached to 7900 Series communications processors are described to the DCP within such a processor in the standard manner for 7900 Series devices. GEORGE is able to access such information, held by the DCP, when the 7900 is introduced to the system. Thus there is no need to issue GEORGE commands to describe the configuration of a 7900. To prevent confusion between local and 7900 peripherals, when GEORGE writes the 7900 information to the I.D.F., any record of a 7900 identifier will be prefixed with the letter I. GEORGE will expect the operator to prefix any input message reference to a 7900 identifier with I.

Using the naming system it is therefore possible to uniquely identify each peripheral on the installation regardless of how such peripherals are connected to the central computer. Notably this scheme ensures that all operator communication contains unique reference to devices; thus if a message about a remote peripheral has to be output at the central operator's console, no confusion will occur.

The Installation Description File (I.D.F.)

The I.D.F. is a direct access file that contains configurational and usage information for character and message buffering communications equipment. GEORGE obtains configuration information about message buffering multiplexers from the resident DCP. However, any device that is used by GEORGE and is attached to character buffering equipment must be described using the **HARDWARE** command, that is, if it is

- 1 A MOP terminal
- 2 A sentinel
- 3 A 7020 being used by the system

There is no need to describe any other device that is attached to a line.

The I.D.F. also contains information on how the hardware is to be used by programs running under GEORGE. It contains the name of conceptual multiplexers, introduced to the file by **CONCEPTUAL** commands, and defines the real character buffering multiplexer lines or message buffering multiplexer identifiers that are attached to each conceptual, this information being provided by **ATTACH** commands. A description of conceptual multiplexers is given on page 42.1.

The I.D.F. also contains information about the peripheral clusters on the installation. These are defined by means of special **CONSOLE** properties and are described on page 30.

SETTING UP THE I.D.F.

When :SYSTEM.IDF does not exist

- 1 When GEORGE has been loaded and the filestore has been set up issue **HARDWARE** commands to define character buffering equipment in detail.
- 2 Issue any **CONCEPTUAL** and **ATTACH** commands, and commands to define peripheral cluster.
- 3 To initiate MOP or system use of 7020s on character buffering devices defined by **HARDWARE** commands (in 1, above), issue **MOP ON** commands for each device.
- 4 Since message buffering multiplexers are set up in a **MOP OFF** state, issue a **MOP ON** command to start the 7900 system running.

The commands issued by the operator to construct a working I.D.F. may be issued individually or by means of pre-defined macros. Once these have been given GEORGE is ready for transmission to or from the 7020s and sends the message:

GEORGE 3 MK m.n READY

followed by the time and date, to all 7023 consoles attached to 7020 terminals. If a 7023 is not attached, or is inoperable, an informatory message is sent to the central operator's console.

When :SYSTEM.IDF does exist

When GEORGE has been loaded and the filestore has been set up a check is made that the character buffering and message buffering peripheral units on the installation correspond with those listed in the file :SYSTEM.IDF. If units exist on the installation that are not listed in the I.D.F., then these units are added to the I.D.F. Any units that are said to exist in the file but do not in fact exist on the installation are not removed from the I.D.F. but simply marked **WRONGed**.

GEORGE will ask if the I.D.F. needs updating for each message buffering multiplexer by outputting the question.

IS A CONFIGURATION UPDATE REQUIRED IN UNIT n?

to the operator's console. Normally there will have been no reconfiguration and the reply will be **NO**.

If a configuration update is required, the operator will reply **YES**. GEORGE will ensure that the unit is **MOPped OFF** and output the message:

UNIT n MOP OFF COMPLETED

if the unit was previously marked as MOPped ON in the I.D.F. GEORGE will then obtain a configuration report from the DCP (Dedicated Control Program) and use it to update the I.D.F.

This action will be completed before the GEORGE READY message is output, that is, before processing can commence.

The only difficulty arises when a character buffering unit that exists on the installation has on it a different number of lines from the number recorded in :SYSTEM.IDF. If this occurs, and/or if a configuration update has been requested, all peripheral clusters declared for the installation are destroyed (that is, all attributions and associations of console properties are cancelled) and the console for each cluster is changed to the central operator's console. Then after the message

GEORGE READY

the following message is output for any character buffering unit which has been found to have the wrong number of lines:

UNIT *n* HAS BEEN RECONSTRUCTED: PLEASE REISSUE IDF COMMANDS

The operator must then reconstruct the I.D.F. for character buffering multiplexers in the manner required (see *When :SYSTEM.IDF does not exist*, page 42).

ALTERING THE I.D.F. DURING RUNNING

Communications processors have the facility of changing lines during the running of GEORGE. If a line becomes unusable for some reason the operator can use the control console on the communications processor to tell the DCP to switch the identifier from the broken line to a working line. This can only be done if a suitable dummy line was specified in the DCP specification tape when it was loaded. The change of line is notified to GEORGE and the I.D.F. is updated accordingly.

The HARDWARE command may be used to update the I.D.F. for a MOPped ON character buffering multiplexer during the running of GEORGE only if the line concerned is either ATTACHed or defined as NULL. If the line was previously defined as NULL GEORGE will attempt to converse with it in the appropriate fashion immediately; if the line was ATTACHed the new line description will be utilised when the attachment is released.

LISTING THE I.D.F.

A listing of the I.D.F. may be produced at any time during the running of GEORGE and is initiated by the operator typing in the command IDFLIST (see Chapter 8). This produces in readable form on a line printer the information currently held in the I.D.F. It is only possible to issue one IDFLIST command at a time; second and subsequent attempts will cause an error reply.

Conceptual multiplexers

WHAT IS A CONCEPTUAL?

The conceptual multiplexer facility is provided in order that a single hardware multiplexer can be split between GEORGE (for MOP and system use of 7020s) and several object programs, each of which is written to control a whole multiplexer. Thus, any object program that has ONLINED a conceptual multiplexer, addresses that conceptual multiplexer as if it were a complete hardware multiplexer, and the hardware boundaries between communications devices disappear.

On a character buffering system it is possible for one or more hardware uniplexers to be included in the definition of a conceptual multiplexer, the other lines, if any, being part of a hardware multiplexer.

In addition, a single line conceptual may be used as a conceptual uniplexer (peripheral type 10), even if it is made up of a single line of a hardware multiplexer. The user specifies whether his program addresses a multiplexer or uniplexer by means of the peripheral type specified in the ONLINE command. An interrogating typewriter (also type 10) cannot be simulated by GEORGE from a single line conceptual, but a program written to use an interrogating typewriter is not detected by GEORGE as being in error; GEORGE simply treats orders to peripheral type 10 as if they were orders to a uniplexer.

SETTING UP CONCEPTUALS

A conceptual is defined by means of a CONCEPTUAL command issued from the central operator's console. This command defines the name of the conceptual. For a character buffering conceptual it also sets a maximum number of lines that may be attached to it. The maximum must be less than 64 and it is not necessary to have real lines attached to each slot in the conceptual when the program is running. For a message buffering conceptual

the **CONCEPTUAL** command also defines the range of conceptual identifier numbers allowed to be attached. The maximum identifier number allowed is 4095, and again it is not necessary to have real identifiers attached to each slot in the conceptual when the program is running.

The **CONCEPTUAL** command does not actually allocate any real character buffering multiplexer or uniplexer lines or message buffering multiplexer identifiers to the conceptual it names. Allocation of real lines or identifiers is done by means of the **ATTACH** command, which may be issued either by the operator or by a MOP terminal user.

The **ATTACH** command may not be issued until the named conceptual has been declared and when the line or identifier to be attached is not in use by the system; the operator may then issue the **ATTACH** command which requests that, in the case of a character buffering multiplexer, a specified line be attached to a named conceptual as one of the conceptual lines, or, in the case of a message buffering conceptual, a specified identifier be attached to a named conceptual as one of the conceptual identifiers. Lines or identifiers are only fully attached to a conceptual after the conceptual has been **ONLINED** to a program. Up to this time the operator can abandon the **ATTACH** by issuing a **CANCEL ATTACH** command.

All lines or identifiers attached to a conceptual are automatically detached when the conceptual is released from a program or when such a program is deleted. For a more detailed discussion of how conceptals may be used on-line to programs, see Chapter 11 of the manual *GEORGE 3 and 4 Operation Management*.

The BROADCAST and SPEAK commands

The **BROADCAST** command enables the central operator to communicate messages to all or a selection of MOP terminals from which user jobs are being run. It is described fully in Chapter 8. This facility is not available to remote operators.

The **SPEAK** command enables an operator (whether central or remote) to send messages to another operator's console. Further details of this command and its use are given in Chapter 8 and also in the section *Inter operator communication* on page 42.5.

Peripheral clusters

A *cluster* is a group of peripherals and MOP terminals, attached to the central processor by communications equipment, that have been associated with a specified operator's console in such a way as to form a mini-installation. Normally a cluster will consist of the peripherals attached to a 7020 terminal plus perhaps some nearby MOP terminals, and the 7023 console on the 7020 terminal will be the *nominated console* for the cluster (sometimes referred to as the cluster console). Other combinations are however possible.

All messages relating to the peripherals and MOP terminals in the cluster are sent to the nominated console for that cluster. The output from all jobs initiated from the cluster will generally be sent to output peripherals in the cluster, but it is possible to specify other arrangements.

Clusters have to be set up by the installation manager issuing certain **PROPERTY**, **ATTRIBUTE** and **ASSOCIATE** commands. Any peripherals and MOP terminals not explicitly clustered are treated by **GEORGE** as central devices.

Basic peripherals that have been grouped into clusters are therefore known as *cluster* peripherals. All other basic peripherals are termed *central* peripherals.

Details of setting up clusters and removing cluster devices from system use are given in Chapter 11 of the manual *GEORGE 3 and 4 Operation Management*.

The use of 7020 remote data terminals

A 7020 terminal may either be used on-line to an object program under **GEORGE**, in which case the object program controls the transmission of data to and from the terminals, or could be used for remote job entry to **GEORGE**, when **GEORGE** controls the transmission of data to and from the terminal. The latter case is known as system use of the 7020 terminal and **GEORGE** aims to make the terminal look like a small installation (see *Peripheral clusters*, above), to the user and the operator. Thus jobs may be entered from the terminal in exactly the same form as jobs entered to **GEORGE** on local peripherals.

Operating clusters

As soon as the necessary commands have been given to define the cluster (of which the 7020 terminal may be part or all) and a MOP **ON** command has been given to start **GEORGE** using the devices (see *Setting up the I.D.F.*, page 42), the following message is sent to the 7023 teletypewriter:

GEORGE 3 MK *m.n* READY

This is followed by the time and date and indicates that GEORGE is ready to receive input or send output to the terminal. It should be noted that the **HARDWARE** command expects a 7023 typewriter on the 7020 terminal and if one is not defined, an error message is given on the central operator's console.

GEORGE starts polling the 7020 terminal and continues to poll at intervals of 4 seconds if the terminal is operable or at intervals of 1 minute if the terminal is inoperable until one or more of the terminal peripherals is allocated.

The interfaces of the 7020 terminal are serviced in the following order:

- 1 Typewriter input
- 2 Typewriter output
- 3 Line printer
- 4 Paper tape punch
- 5 Card reader
- 6 Paper tape reader

Thus, if both the card reader and paper tape reader are loaded with input and allocated, the cards will be read first. It is possible to have more than one data stream in progress on one terminal because if a hold up occurs in one stream then another one will be started. All messages relating to the terminal peripherals and MOP consoles belonging to the same cluster will be sent to the nominated console.

The peripherals on a 7020 terminal are referred to in system messages from GEORGE using their unique alphanumeric identifier (see the *HARDWARE command* page 100), for example

PLEASE ENGAGE UNIT IAA1

In commands from the nominated console, the operator must use the form *Iaan* when referring to terminal peripherals, where *aa* is the terminal identifier and *n* is the relevant interface number. If the nominated console breaks down, GEORGE will route messages concerning the relevant cluster devices to the central operator who can help by communicating with the remote operator (see *Inter operator communications*, page 42.5).

OPERATING A REMOTE CONSOLE

To input a message the operator should press the INPUT button and wait for the INPUT light to switch itself on, indicating 'type now'. The operator should then type the message and press Accept. GEORGE will then read this message and to indicate that the message has been read a newline will be output.

To input a group of messages the operator should press the INPUT button immediately after pressing Accept. Then he should type a newline followed by the message and Accept.

OPERATING REMOTE BULK PERIPHERALS

The operator is allowed to load bulk data on the input devices but it is recommended that commands should be input singly (except in the case of RUNJOBS). This is to allow the operator to keep track of the data and thus be able to perform a total restart should this be necessary.

REMOTE JOB ENTRY

Jobs are entered from 7020 installations in the same way as they are entered centrally. They may however give different effects depending on whether they are designated central or cluster. (It should be noted that according to the definitions of central and cluster peripherals, a MOP terminal which is not part of a cluster has the designation central, even though it may be physically remote from the central installation.)

A job is designated a cluster job if the following conditions are satisfied:

- 1 The job description commencing with an INPUT command is read in from a central or cluster peripheral and subsequently run by inputting a RUNJOB command at the cluster console or by issuing a RUNJOB command from another cluster job.
- 2 The job description commencing with a JOB command is read in from a cluster peripheral.
- 3 A LOGIN command is issued from a cluster MOP terminal.

A job is designated a central job if the following conditions are satisfied:

- 1 The job description commencing with an INPUT command is read in from a central or cluster peripheral and subsequently run by inputting a RUNJOB command from the central operator's console or a central job.
- 2 The job description commencing with a JOB command is read in from a central peripheral.
- 3 A LOGIN command is issued from a central MOP terminal.
- 4 A cluster job after issuing a CENTRAL command.

If JOBLIMIT for the installation is reached when a JOB or RUNJOB command is issued from a cluster, the message

JOBLIMIT REACHED

is sent to the cluster console. The card reader or tape reader will remain allocated, but will stop. Once the number of jobs in the system is reduced, the job will restart normally without any operator action.

JOB and RUNJOB commands entered centrally do not take precedence over jobs entered remotely, or vice versa, but are started in the order in which they are accepted by the system.

If a command error occurs in NO USER context on a 7020 input device, an error message is output to the nominated console and the device is WRONGed. The operator must take some action to reset the input device and then RIGHT the device to make it available to GEORGE.

RE-ROUTING LISTFILES

It is possible to output files destined for a remote device in one cluster to a device in another cluster by the operator issuing a CHANGELIST command. For example, if a file OUTPUT is destined for a line printer in the LONDON cluster, on special stationery INVOICES, the original command would be:

LISTFILE OUTPUT, *LP,PROPERTY LONDON&INVOICES

(Note the format used to specify more than one property in the command.)

If the printer were inoperable, the listing could be re-directed to the CARDIFF cluster by the command:

CHANGELIST username.jobname,*LP,OUTPUT,PROPERTY CARDIFF&INVOICES

If the listing is to be redirected centrally, the property CENTRAL is assumed by default. The CHANGELIST command would then be:

CHANGELIST username.jobname,*LP,OUTPUT,PROPERTY INVOICES

It is also possible to stop the output from a LISTFILE before it has begun by using a STOPLIST command (see Chapter 8). The TERMINATE command may be used to stop a listing that has already begun.

COMMANDS AVAILABLE TO THE REMOTE OPERATOR

The commands which are available from a remote operator's console are as follows:

ABANDON (AB)

ATTRIBUTE (AU) and CANCEL ATTRIBUTE

CANTDO (CD)

CHANGELIST (CH)

DEAD (DD)

NEW (NE)

PAGES (PA)

REDON (RO)

RETRIEVE (RV) and CANCEL RETRIEVE

RIGHT (RI)

RUNJOB (RJ)

SPEAK (SK)

STOPLIST (SL)
 TERMINATE (TE)
 UNLOAD (UL)
 WHATJOBS (WJ)
 WHATLIST (WL)
 WHATSTATE (WS)
 WRONG (WR)

The commands ATTACH and CANCEL ATTACH are available from a remote operator's console on a message buffering system only.

All other built-in operator commands will generate a context error if issued from a remote console.

Command errors in NO USER context are reported to the console of the originating cluster; any other command errors in job descriptions are reported to the monitoring file in the usual way.

INTER OPERATOR COMMUNICATION

If the central operator wants to send a message to one or all of the remote operators' consoles, he can do so by issuing a SPEAK command (see Chapter 8). In the same way, a remote operator can send a message to the central operator's console or to the operator's console of another cluster. This means therefore that operators can converse with one another via their consoles.

Inoperable MOP terminals and jobs in limbo

If a MOP terminal becomes inoperable GEORGE takes the following action:

- 1 A message is sent to the operator's console. The form of the message depends on the way the terminal is connected, as follows:
 - (a) If it is connected through a character buffering multiplexer the message is:
 UNIT *number* LINE *number* HAS BECOME INOPERABLE
 - (b) If it is connected through a message buffering multiplexer the message is:
 UNIT *number* IDENTIFIER *number* HAS BECOME INOPERABLE
 - (c) If it is connected through a uniplexer the message is:
 UNIT *number* HAS BECOME INOPERABLE
- 2 If a job is being run from the terminal it will be stopped and put into a state of limbo.

LIMBO

If a MOP terminal becomes inoperable the job being run will remain in limbo (that is, dormant in the machine) for the time specified by the WAITTIME installation parameter or until the job is CONNECTed by the user or ABANDONed by the operator. Such a job will waste core and prevent another MOP job being started if the system is fully loaded according to the appropriate installation parameters. If a communications device breaks down there may be several jobs in this state and the operator must be aware of the action to take according to the circumstances. All the jobs in limbo may be abandoned by a temporary reduction of WAITTIME to its minimum value (with consequent danger of abandoning background jobs waiting to be connected).

EXOTIC PERIPHERALS

In GEORGE, an exotic peripheral is one which has the following characteristics:

- 1 Its peripheral type is other than 0, 1, 2, 3, 4, 5, 6, 9, 10, 11, 13, 24, 25, 26, 28, 50, 51, 60 or 63
- 2 There is a GEORGE Executive package available for it
- 3 It cannot be shared by programs

Exotics may be grouped into three basic classes: ordinary engageable, ordinary non-engageable or flag-setting. An ordinary engageable exotic is one which can be dis-engaged by extracode and engaged by the operator. This is not

true for a non-engageable exotic or a flag-setting exotic.

GEORGE will treat exotics in a different way from standard GEORGE peripherals; among the differences are the following:

- 1 An exotic will never be used by GEORGE for any system purpose; this implies that exotics must always be connected ONLINE to an object program
- 2 Ordinary input exotics will not respond to the pressing of an ENGAGE button to read a command. Thus DOCUMENT, INPUT, JOB and RUNJOB commands are not catered for on such devices and the REDON command is not applicable
- 3 LISTFILE commands cannot be performed on exotic output devices, and ONLINE output will not have the standard document headers

When an ordinary exotic peripheral is ONLINED to an object program, GEORGE will disengage the device and output the following message to the operator's console

PLEASE CHECK *data description* ON UNIT *n*

where *data description* is the second parameter of the ONLINE command for the device. The operator should then make sure the device is in the correct state and indicate to GEORGE that the device is ready for use by pressing the ENGAGE button. If the operator is unable to carry out the request, he may issue a CANTDO command: this will generate an error which will be dealt with by the job description.

If the device is a non-engageable exotic, the message

PLEASE CHECK *data description* ON UNIT *n*

will still be output but GEORGE will not wait for an engage button and the device will be made on-line to the program without any pause for operator intervention.

For flag-setting exotics GEORGE assumes that the object program controlling the device will handle any necessary conversations itself.

INTER PROGRAM COMMUNICATION (IPC)

IPC is a facility whereby one program can transmit data to another program either in the same machine or in another machine connected by an Inter Processor Buffer (IPB). The connection between two programs is known as an *IPC link*.

Operational aspects of IPC

If a 7210/0 Inter-Processor Buffer (IPB) is inoperable and GEORGE wishes to use it, the message:

PLEASE ENGAGE UNIT *z*

will be output on the operator's console, and repeated at two-minute intervals until either of the following occurs:

- 1 The device is engaged. (This would normally be done by switching on the power supply and/or moving the control switch from the "offline" to the "online" position).

or

- 2 A corresponding CANTDO command is given. In this case all IPC links open through the device will be disconnected (that is, disestablished) and no further dialogue will be attempted unless and until the device becomes operable again.

Note that an IPB may appear inoperable to GEORGE and a PLEASE ENGAGE UNIT *z* message may be output when the IPB itself is online but the remote processor has gone down.

If the number of errors in the dialogue transmitted through an IPB reaches an unacceptable level (for example, if one machine has gone down for either hardware or software reasons), all links open through the IPB will be disconnected and all dialogue suspended. The message

DIALOGUE THROUGH IPB *z* SUSPENDED

will be output on the operator's console.

Dialogue will only be resumed when the processor at the other end of the link goes through the initialisation routine, which it will normally do automatically when GEORGE is loaded. When dialogue is resumed, the message

DIALOGUE THROUGH IPB z RESUMED

will be output.

IPC and the property system

If more than two processors are linked together each Inter Processor Buffer can be given a property so that, when setting up an IPC link, a user can specify which processor he wants to be linked to by requesting the relevant property.

Properties for IPC are ATTRIBUTED to Inter Processor Buffers in the same way as properties for basic peripherals. The effects of declaring the properties to be PERMANENT/TEMPORARY or EXCLUSIVE/INCLUSIVE are also the same as for basic peripherals and therefore if a user requests an IPB with a TEMPORARY property that is not currently attributed, GEORGE will ask the operator to ATTRIBUTE it. However, the operator should try to avoid disturbing any process already using the IPB.

Chapter 4 The back-up system

The GEORGE back-up system provides two main facilities:

- 1 It enables GEORGE to be started, or restarted after loss or corruption of direct access backing store, with a recent and usable version of the filestore.
- 2 It enables the filestore to occupy more space than is available on direct access backing store and helps clear backing store jams.

The system consists of six routines which may be grouped into three categories:

- 1 **START ROUTINES** These routines, known as *Backmap* and *general restore*, are concerned with the starting or restarting of GEORGE.
- 2 **MAINTENANCE ROUTINES** These routines, known as the *incremental dumper*, the *backing store unjammer* and the *file retrieval system*, are concerned with maintaining filestore resilience, clearing jams and extending the filestore.
- 3 **DUMP TAPE PROCESSING** These routines are concerned with limiting the number of tapes holding dumped increments. A choice is offered between the tape-to-tape processor (TTTP) and the built-in dump tape processor, but the use of the former is recommended.

THE START ROUTINES

Backmap

When GEORGE is started or restarted after a breakdown, the E.M.S. routine, after ascertaining what backing store space is available to GEORGE, enters the routine Backmap. When Backmap is entered, it finds out if a general restore is required by the message:

RESTORE?

Unless it is definitely known that the filestore is corrupt, the operator is advised always to type NO (or N). There are two specific cases, however, where a reply of NO will not be accepted. These are:

- 1 When the filestore was created by a mark of GEORGE different from that currently being loaded.
- 2 When the system has broken down during a previous general restore.

In these two cases, or if the operator has typed YES (or Y), Backmap enters the *general restore routine* (see below). Alternatively, instead of typing YES, the operator may type the number that would have been given in reply to the question INCREMENT? (See *General restore routine* below).

If a general restore is not required, Backmap checks that a valid version of the filestore is present on backing store. (An invalid filestore is one with a directory or vital system file on an unobtainable residence.) If the filestore is found to be invalid the message:

PLEASE LOAD RESIDENCE *n* OR ASK FOR A GENERAL RESTORE

is output. If the operator is unable to load the required residence he should initiate a general restore, otherwise he should load the required residence and follow the normal start procedure.

Both Backmap and the general restore routine are entered only at E.M.S., and are protected from interaction with other routines.

SYSTEM ERRORS DURING BACKMAP

If the system breaks down during Backmap it must be decided whether to do a general restore or to try again. If the breakdown was caused by a hardware error (for example SYSTEM ERROR B GEOERR TRANFAIL or a processor

failure) there is a good chance that a further attempt will succeed. If the breakdown was due to corruption of filestore then it is unlikely to be worthwhile trying again and a general restore should be initiated. If a second attempt is made and the same error occurs again then it is almost certain that it is due to filestore corruption and a general restore should be initiated.

General restore routine

When the general restore routine is entered by Backmap, it outputs the message:

INCREMENT?

The operator must reply with the number of the last increment which the dumper has specified as suitable (see *The Incremental Dumper*, page 47) and must also engage a deck holding a tape with a copy of the relevant increment on it. If the number typed in is not an integer between 1 and 8388607 inclusive, the error message:

INCORRECT INCREMENT DUMP NUMBER FORMAT

is output and the request for the number is repeated.

If a deck is not engaged, the general restore requests:

PLEASE ENGAGE DECK HOLDING INCREMENT DUMP NUMBER *n*

It is possible that the operator is unable to engage the deck. If so he replies with a CANTDO command. In this case the message:

INCREMENT?

is output again, and the operator should reply with the number of a suitable increment that he can engage. If the operator types any command other than a CANTDO (or CD) command, the message

COMMAND NOT RECOGNISED

will be output.

When a dump tape is engaged, the following message is output on the operator's console:

DUMP TAPE *z* IS ENGAGED

If the tape obtained is not a dump tape, the error message:

z NOT INCREMENT DUMP TAPE

is output.

Having obtained a dump tape, the general restore routine checks the tape name and increment number against the information supplied by the operator. If the specified increment is not on the tape, the routine outputs the message:

INCREMENT NO *n* NOT ON TSN *z*

When the correct tape and increment have been found, the routine begins to restore the dump of the filestore to backingstore.

If it is found after reading the specified increment that it is not suitable for a general restore, the following error message is output:

z NOT SUITABLE FOR RESTORING FILESTORE

After the general restore, control returns to Backmap.

TAPE FAILS AND FORMAT ERRORS

If the general restore routine encounters a tape fail or a format error on a dump tape, a message of the form

TAPE FAILURE IN INC. NO. *n* ON IDT *z*

or

FORMAT ERROR IN INC. NO. *n* ON IDT *z*

will be output, followed by the message:

LAST SSF SENTINEL READ *filename*

The last message provides an indication of where on the tape the failure or format error has occurred.

When reading a dump tape, the general restore routine matches SSF sentinels with ESF sentinels. If one of these is found to be missing or incorrect, one of the following messages will be output:

SSF SENTINEL WAS EXPECTED

ESF SENTINEL WAS EXPECTED

ESF OR SSF SENTINEL WAS EXPECTED
NAME IN SSF SENTINEL NOT NAME IN TOP FCB
BLKS READ FROM MT NOT SAME AS IN SSF SENTINEL
LONG BLOCK READ

If an increment is genuinely corrupt, any of the following set of messages may be output:

NAME RECORD IN A DIRECTORY IS INCORRECT LENGTH
A TRAPS RECORD WAS EXPECTED
A NAME RECORD WAS EXPECTED
AN INCOMPLETE DIRECTORY ENTRY ENCOUNTERED

These messages, together with the one stating the position on the tape, should provide an indication of the cause of the failure.

After the appropriate messages have been output, the general restore routine deletes everything it had done up to the time of encountering the failure or format error and then asks again for the increment number by outputting the message:

INCREMENT?

When this occurs, the operator should try (in order) to:

- 1 Reload the same tape on a different deck and type in the same number.
- 2 Load another tape containing the same increment number.
- 3 Load a tape containing a different increment number and type in the new number.

There are some circumstances where a failure may cause GEORGE to break and enter the postmortem routine. If this occurs, one of the following explanatory messages will be output:

MT *n* KNOWN TO S/SERIAL BUT NOT TO S/INCINDEX
NO RECORD IN S/INCINDEX OF INCREMENT *n*
NO RECORD IN S/SERIAL OF COPY TAPE
NO RECORD IN S/DUMPER OF COPY TAPE
NO RECORD IN S/INCINDEX OF MT *n*
UNABLE TO OPEN S/DUMPER
UNABLE TO OPEN S/INCINDEX
UNABLE TO OPEN S/SERIAL

Juggernaut Restorer

When a general restore has been done from the latest increment, it is likely that many of the most frequently used files in the filestore (for example, system macros) will not be on-line. These files will therefore have to be RETRIEVED from the dump tapes before they can be used. To perform this retrieval efficiently the Juggernaut job is automatically run after a general restore. This job may request the operator to load dump tapes containing those files that the manager considers should always be on-line when sufficient space is available. The job description in :MANAGER JUGGERNAUT can be edited to perform other actions which are thought advisable after general restore.

The :MACROS.EMSJOBS facility

This facility is provided to enable users to run certain jobs each time GEORGE is restarted. If the file :MACROS.EMSJOBS exists, it will be executed at EMS time. Jobs can then be automatically initiated under various users by placing suitable RUNJOB commands in this file.

Dump checking

After GEORGE is restarted it is possible that some increments will be unavailable because all the tapes that hold them have been TAPEWRONGED when GEORGE was last running. This means that any files on those tapes will

appear to be lost. To prevent this happening unnecessarily, if there are any unavailable increments at the end of Early Morning Start, a job (:DUMPER.Q) is initiated to remind the operator about such increments.

The job outputs the following message:

MTS *a,b,c*, ARE WRONGED, MAKING ALL FILES ON INCREMENTS *d,e,f* UNAVAILABLE

The operator is then asked:

CAN YOU TAPERIGHT ANY OF THESE TAPES?

If the operator replies N, then users requesting files in these increments will be told they are not available.

If the operator replies Y, the following message is output:

PLEASE DO TAPERIGHTS

The job waits 2 minutes and then checks all increments again. When all increments have been checked satisfactorily the job ends.

THE MAINTENANCE ROUTINES

The backing store unjammer

The backing store unjammer is called in when the proportion of backing store exceeds the value of the installation parameter BACKJAM.

When the unjammer is called in it will output the following message:

BACKING STORE UNJAMMER STARTED ON *date* AT *time* (GEORGE TIME *time*)

where GEORGE TIME *time* refers to GEORGE Mean Time (see *Early Morning Start routine*, page 22).

Each new residence that is detected as jammed will be identified to the operator, at the beginning of the unjammer's pass, by the following message:

JAM ON RESIDENCES $n_1 n_2 n_3 \dots$

As the unjammer scans the filestore it will indicate those residences that are above threshold (determined by the installation parameter BACKTHRESH), but below the level of BACKJAM, by outputting the following message:

ABOVE THRESHOLD ON RESIDENCES $n_1 n_2 n_3 \dots$

If a jam is detected in a special residence (one on which only ALLOCATED files may reside), the unjammer will not attempt to discard files; instead the operator will receive the message:

CANCEL ALLOCATE FILES FROM RESIDENCES $n_1 n_2 n_3 \dots$

If the unjammer calls in the dumper, the following message is output:

BACKING STORE JAM:— DUMPER STARTED AT *time*

Should the unjammer detect the condition where it is unable to clear a jam, even after calling in the dumper, it will output the following message:

UNJAMMER STUCK:— PLEASE TRY ABANDONING JOBS (BUT NOT THOSE WAITING FOR BACKING STORE) OR RELOAD THE SYSTEM

At the end of a pass through the filestore tree, the unjammer will identify those jammed residences cleared below BACKJAM and any other residences cleared below threshold with the following messages:

JAM CLEARED ON RESIDENCES $n_1 n_2 n_3 \dots$

RESIDENCES $n_1 n_2 n_3 \dots$ CLEARED BELOW THRESHOLD

When no further residences are jammed, the following message is output:

BACKING STORE JAM CLEARED AT *time*

It is possible that, during the time the unjammer is at work, other residences that were not jammed become filled above their threshold. The unjammer will indicate this condition if it occurs at the end of a jam by outputting the following message:

RESIDENCES $n_1 n_2 n_3 \dots$ NOW ABOVE THRESHOLD

This message may also apply to residences which were jammed and which the unjammer has cleared below the jam limit but not below the threshold limit. (This is because a jam is considered cleared when all residences are below the jam limit but not necessarily below the threshold.)

While the unjammer is running it may find it necessary to alter the value of the installation parameter FORMULA. However if the new value of FORMULA is out of range then the following message will be output:

BACKING STORE JAM:— UNREASONABLE LIMIT ON IP FORMULA — PLEASE ALTER TO
INCLUDE VALUE *m*

The incremental dumper

The incremental dumper dumps to magnetic tape copies of all files marked as requiring dumping. The frequency of dumps is controlled by the installation parameter DUMPTIME. (Installation parameters are described in Chapter 2.1 and are listed in the specification of the INSTPARA command in Chapter 8). The DUMPTIME installation parameter can be overridden by the operator by using a DUMP command. (DUMP TOTAL will cause the entire on-line filestore to be dumped except for files open for writing or not on-line but this facility should be used with extreme caution since dumping of the entire on-line filestore takes a long time and repeated use of DUMP TOTAL is therefore inefficient). The DUMPTAPES installation parameter specifies the number of tape copies of the dump. (This facility provides greater filestore resilience).

The incremental dumper may also be called in automatically by the backing store unjammer. Note that the unjammer is called in when the proportion of backing store exceeds the value of the installation parameter BACKJAM, but the unjammer may or may not call in the dumper.

When the dumper is entered, it will aim to dump on the tapes that it used for the last increment. If, however, it requires extra tapes it will look for the first available unused tape called GEO3DUMPTAPE in the directory of :DUMPER (see *Dump tape releasing*, page 49). If none such exist, it will inform the operator with the message:

DUMPER REQUIRES POOL TAPE

and then request a tape from GEORGE's pool of scratch tapes as follows:

PLEASE LOAD POOL TAPE ON UNIT *a b c d*

When the dumper has found the required number of tapes, it outputs the message:

DUMPER INTENDS TO DUMP ON MTS *list of tape serial numbers*

If any required tape is not loaded, the following message is output:

PLEASE LOAD MT *tape serial number* WPR ON UNIT *a b c d*

The operator should normally load the tapes as requested, but if for any reason he cannot obey either of the requests to load, he should issue a CANTDO command. The dumper will then output the message:

DUMPER :MT *tape serial number* NOT USED DUE TO UNAVAILABILITY

If one of the tapes used for the last increment has given read failures, the dumper will not reuse that tape and will not request it, but will output the message:

DUMPER :MT *tape serial number* NOT USED DUE TO FAILURE HISTORY

Either of the last two messages above will be followed by the question:

:DUMPER,*jobname* QUESTION:

DO YOU STILL WISH TO DUMP ON *n* MTS?

where *n* is the number of dump tapes it is currently trying to dump on, and
jobname is of the form *An* where *n* is an integer that is increased by 1 each time a general restoreable increment is produced.

The operator can reply either

ANSWER :DUMPER,*jobname*,YES

or

ANSWER :DUMPER,*jobname*,NO

If he answers YES, the dumper will ask for another tape, from the pool if necessary, to replace the tape which the operator could not load. If he answers NO, the dumper will assume that one tape less is to be used for this

increment. If this means that there are in fact no tapes left on which to dump, the message

INCREMENT *n* ABANDONED AND OBLITERATED

is output, and no dump is made until the next DUMPTIME interval has elapsed.

When all the dump tapes have been positioned, the dumper outputs the message:

START OF INCREMENT *n* ON MTS *tape serial numbers*

and dumping of the increment will begin.

The dumper then searches the whole on-line filestore dumping all files requiring dumping, except those that are open for writing.

When the dumper has finished its search, it outputs the message:

END OF INCREMENT *n* ON MTS *tape serial numbers* – COMPLETE

and also the message:

INCREMENT *n* on MTS *tape serial numbers* SUITABLE FOR GENERAL RESTORE

Only such an increment may be used for a general restore.

The operator should note, for each increment said to be COMPLETE, the increment number, tapes to which it was dumped and whether it is 'SUITABLE FOR GENERAL RESTORE', so that he will know the tape serial numbers of the tapes containing the latest incremental dump suitable for a general restore.

Example

DUMPER INTENDS TO DUMP ON MTS 350015 350016
14.35.00

PLEASE LOAD MT 350015 WPR ON UNIT 24 25 26 28
PLEASE LOAD MT 350016 WPR ON UNIT 30 25 26 28
CD LOAD MT 350015 WPR

14.36.00

:DUMPER,A2 QUESTION: DO YOU STILL WISH TO DUMP ON 2 MTS?

AN :DUMPER,A2,YES

CD LOAD MT 350016 WPR

14.37.00

PLEASE LOAD TAPE FROM POOL ON UNIT 30 25 26 28
MT 360137 IS NO LONGER A POOL TAPE

:DUMPER,A2 QUESTION: DO YOU STILL WISH TO DUMP ON 2 MTS?

AN :DUMPER,A2,YES

MT 350136 IS NO LONGER A POOL TAPE

DUMPER INTENDS TO DUMP ON MTS 350137 350136

START OF INCREMENT 1504 ON MTS 350136 350137

END OF INCREMENT 1504 ON MTS 350136 350137

INCREMENT 1504 ON MTS 350136 350137 SUITABLE FOR GENERAL RESTORE

14.55.00

If while writing to the tapes the dumper has a write fail or reaches an end-of-tape marker on the tape, it outputs the message:

DUMPER: MT *tape serial number* WRITE FAIL

or

DUMPER: MT *tape serial number* END-OF-TAPE

and proceeds to complete the increment on all tapes up to the write fail or end-of-tape. If for any reason the dumper cannot complete the increment (for example, a readfail occurs while it is reading backwards on one of the tapes) it outputs the message:

END OF INCREMENT *n* ON MTS *tape serial numbers* – INCOMPLETE

Such an INCOMPLETE increment will never be used for retrieving and therefore does not need to be noted by the operator.

If the dump was completed successfully, the following message is output:

END OF INCREMENT *n* ON MTS *tape serial numbers* — COMPLETE

but does not indicate that the increment is suitable for a general restore. It then repeats the whole procedure, requesting a new set of tapes and commencing the filestore search where it left off.

CANCELLING A DUMP

If a dump is initiated and the operator should then wish to cancel it, the dumper routine can be terminated by issuing CANTDO commands in the following manner:

```
DUMP TOTAL
DUMPER INTENDS TO DUMP ON MT 360254
PLEASE LOAD MT 360254 WPR ON UNIT 20 21 22 23
CD LOAD
:DUMPER,A QUESTION: DO YOU STILL WISH TO DUMP ON 1 MT?
AN :DUMPER,A,N
INCREMENT 21 ABANDONED AND OBLITERATED
```

THE DUMP TOTAL FACILITY

This facility is usually used to make a total dump of the filestore after it has initially been created from an incremental dump. If used, it should be used with caution as it dumps all files that are on-line, but which are not open for writing.

DUMP TAPE RELEASING

This section applies only if the built-in dump tape processor is used. If the tape-to-tape processor is used, dump tape releasing will be performed automatically.

A tape that contains only obsolete increments may be released so that it will be available for use again, either as a dump tape or as a pool tape. The DUMP command can be used to return tapes to the pool, or to mark dump tapes for reuse by the dumper.

To return obsolete tapes to the pool, the operator should issue a DUMP RETURN command (see page 93). The following message will be output to the operator's console:

```
DUMPER: MT tape serial numbers RETURNED—INCREMENTS OBSOLETE
```

Alternatively, the DUMP REUSE command may be used. This will cause obsolete dump tapes to be marked 'for reuse by dumper' instead of returning them to the pool. The message:

```
DUMPER: MT tape serial numbers TO BE REUSED—INCREMENTS OBSOLETE
```

will be output to the operator's console. When the dumper requires a new tape, it will reuse tapes marked in this way and will only ask for a pool tape when no more free dump tapes are available.

If, after dump processing, a DUMP RETURN or DUMP REUSE command is issued, and then it is necessary to go back to a previous dump, difficulties may arise when attempting to use the tapes which were processed and released. It is therefore important not to RETURN or REUSE tapes until it is certain that they will not need to be used again. For example, tapes must not be RETURNed or REUSEd unless at least one increment suitable for a general restore has been produced since they were PROCESSed.

The file retrieval system

From time to time user's jobs will need files that are not on-line and so have to be read from tape. All requests for such files are co-ordinated by the retrieval system which will put the requests on a queue ordered according to the order of the files on the tapes, and open tapes appropriately.

When a tape that is not loaded is required the operator is requested to load it (see *Loading magnetic tapes*, page 32).

If a dump tape is not available to the installation at a particular time, or consistently fails, it may be wronged with the TAPEWRONG command. However, if all the tapes that hold a particular increment are wronged, the user's file will be reported as unavailable.

DUMP TAPE PROCESSING

Two methods of keeping the number of tapes containing incremental dumps to a reasonable figure are available: the built-in dump tape processor and the tape-to-tape processor (TTTP); but the use of the latter is recommended.

The dump tape processor

This routine aims to keep the number of tapes containing incremental dumps to a reasonable figure by searching the tapes on which the oldest unprocessed reliable increment is dumped.

When a tape has been read, all increments on it are recorded in a system file as being 'processed'.

If, on completion of an increment, the dumper finds that all the files dumped in processed increments have been dumped again on other increments, it marks these processed increments as obsolete, and outputs for each one the message:

INCREMENT *n* ON MTS *tape serial numbers* OBSOLETE

The dump tape processor is initiated by the command DUMP PROCESS. It first selects the earliest unprocessed increment and checks to see if it can be processed. An increment cannot be processed unless there is a later increment which is reliable and on available tapes; nor can it be processed if it is on the current set of dump tapes. In these cases, the processor will stop soon after it has output one of the following messages:

INCREMENT *n* NOT PROCESSED BECAUSE IT IS ON CURRENT DUMP TAPES

or

INCREMENT *n* NOT PROCESSED BECAUSE THERE ARE NO LATER RELIABLE INCREMENTS
ON AVAILABLE TAPES

If the increment can be processed, the following message is output:

PROCESSOR STARTING ON INCREMENT *n*

PLEASE LOAD MT *tsn* ON UNIT *m m m*

The processor then tries to open a tape containing the increment. If there are no tapes available (that is, they have been CANTDOed or TAPEWRONGed), the following message is output:

INCREMENT *n* CANNOT BE PROCESSED BECAUSE MTS *tape serial numbers* ARE UNAVAILABLE.
IF THE TAPE(S) WILL NEVER BE AVAILABLE, INCREMENT WILL BE MARKED AS PROCESSED
AND COPIES OF FILES IT CONTAINS WILL BE LOST.

This is followed by the question:

:DUMPER,PROC QUESTION:WILL THE TAPES EVER BE AVAILABLE?

If the operator replies NO, the increment will be marked as processed and the processor will try to process another increment.

It is important to note that a reply of NO does mean that the entire increment is lost.

If the operator replies YES, the increment will not be marked as processed, so that when a tape containing the increment becomes available, it will be processed after a subsequent DUMP PROCESS command.

If a tape is loaded and an increment successfully processed, the following message is output:

INCREMENT *n* PROCESSED

The operator will then be asked

DO YOU WANT TO CONTINUE PROCESSING?

The operator may reply YES or NO. If the answer is YES, the next increment on the tape is processed. If the answer is NO, the processor terminates and the message

PROCESSOR TERMINATED

is output on the console.

If a tape fails while being processed, it is unloaded and requested again. The operator can then either reload it on another deck, or TAPEWRONG it in which case the processor will attempt to process the increment by asking the operator to load another tape containing a copy of the increment.

Example

DUMP PROCESS
PROCESSOR STARTING ON INCREMENT 11
PLEASE LOAD MT 360641 ON UNIT 20 21 22 23

PROCESSOR STARTING ON INCREMENT 11
 INCREMENT 11 PROCESSED
 :DUMPER,PROC QUESTION: DO YOU WANT TO CONTINUE PROCESSING?
 AN :DUMPER,PROC,Y
 PROCESSOR STARTING ON INCREMENT 12
 PLEASE LOAD MT 360665 ON UNIT 20 21 22 23
 CD L
 INCREMENT 12 CANNOT BE PROCESSED AS MT 360665 IS UNAVAILABLE
 IF THE TAPE(S) WILL NEVER BE AVAILABLE,INCREMENT WILL BE
 MARKED AS PROCESSED AND COPIES OF FILES IT CONTAINS WILL BE
 LOST
 :DUMPER,PROC QUESTION: WILL A COPY BE AVAILABLE LATER?
 AN :DUMPER,PROC,N
 INCREMENT 12 HAS BEEN PROCESSED
 :DUMPER,PROC QUESTION: DO YOU WANT TO CONTINUE PROCESSING?
 AN :DUMPER,PROC,Y
 INCREMENT 13 NOT PROCESSED AS IT IS ON CURRENT DUMP TAPES
 DUMP

The tape-to-tape processor (TTTP)¹

TTTP processes dump tapes by copying increments directly from one tape to another, leaving out any files (including directories) that have been erased or re-dumped since the increment was created. Increments are therefore compressed into a smaller space on fewer tapes. Output tapes from TTTP are given the name GEO3PROCTAPE and will be used by the retrieval system wherever possible instead of GEO3DUMPTAPES, thereby providing faster retrieval of off-line files.

TTTP provides a fundamentally different type of processing to that available with the built-in dump tape processor. In particular:

- 1 TTTP does not normally bring old files on-line to be redumped. Thus a lot of unjamming and redumping is avoided, together with the consequent degradation of throughput.
- 2 An increment being processed is not removed from INCINDEX unless it is found to contain no useful files. Copies of the increment are produced in a more compact form on new tapes.
- 3 Up to four different sets of increment tapes can be kept. (The number is controlled by the DEPTH parameter). If all tapes in the latest set are found to be unusable, tapes from older sets are available.
- 4 The off-line filestore is stored in a compact form on the set of most recently processed tapes.
- 5 'Old' files are not mixed in with new ones, so dump tapes are separated into those containing new files which are more likely to be retrieved and those containing old files which can be stored away for long periods. This provides a cheap, well-organized archive store.
- 6 Newer increments can be processed more frequently than older ones, reflecting the greater initial rate of decay of data and the greater need for compactness of useful data on recent, frequently-used dump tapes.

GEO3PROCTAPES are suitable for RETRIEVEs or further processing by TTTP or the dump tape processor, but they cannot be used for a general restore because they do not contain directories and vital system files. For this reason TTTP will not process increments that are less than a specified number of increments old.

Since increments are only normally erased by TTTP when they contain no useful files, INCINDEX may become rather large. To enable old increments that contain only a few useful files to be eliminated, a "re-dump" facility is provided with TTTP. The "re-dump" facility is similar to the dump tape processor in that it causes files to be brought on-line and then re-dumped in a new increment. However, TTTP does this over a longer time-scale than the dump tape processor; files are not brought on-line immediately but are RETRIEVED at the time of the next dump and remain in the filestore until the dump after that. Unlike the dump tape processor, TTTP only brings files on-line when specifically requested to do so by means of the re-dump facility, and the user can therefore avoid core jams and backing store jams by requesting that only small increments be redumped.

TTTP is no less secure than the dump tape processor, and in one way provides greater security in that it allows processed copies of increments to be retained. The dump tape processor does not produce any new copies of an increment.

¹TTTP is based on software developed by the University of Oxford, and LISTDUMP (see page 52) is based on software developed by the University of Hull. ICL wishes to acknowledge the valuable assistance of both universities.

THE OPERATION OF THE TAPE-TO-TAPE PROCESSOR

TTTP is a subject program run by a system macro of the form:

TTTP parameter list

A full specification of the macro and its parameters is given in Chapter 13 of the manual *GEORGE 3 and 4 Operation Management*.

TTTP may be run as a background job, from a MOP terminal or from the operator's console. In all three cases, the username must be :DUMPER and the jobname must be PROC. If the macro is issued from the operator's console, a RUNJOB command is issued for this username and jobname. In this case, (or if it is being run as a background job) TTTP will first output the message:

TTTPROCESSING UNDER OPERATOR'S CONTROL

on the operator's console.

TTTP requires as data information on the current state of the filestore and this information is obtained by reading files prepared by the program LISTDUMP (see the specification of the LISTDUMP system macro in Chapter 13 of the manual *GEORGE 3 & 4 Operation Management*). It is not always necessary to run LISTDUMP before TTTP is run if it has already been run recently. However, TTTP will check whether the information provided by the last run of LISTDUMP is out of date and, if it is, the message:

LISTDUMP MUST BE RUN IN ORDER TO PROCEED

will be displayed on the operator's console.

Input Tapes

TTTP reads :SYSTEM.INCINDEX in order to find possible input tapes for an increment it is about to process. From these, it chooses a tape which it thinks is best, according to a fixed algorithm. Basically, it attempts to find that tape which is non-failing for the increment and which contains most increments and has been produced most recently. If no non-failing tapes are available, TTTP uses tapes which the retrieval system has found to be failing. If even these are unavailable and only tapes found to be failing by TTTP itself are left, the increment is marked to be redumped and TTTP goes on to the next increment to be processed.

If no tapes at all were available for an increment, and the operator has typed CANTDO to all of TTTP's requests for tapes, TTTP will output the message:

HAVE ANY TAPES NOW BEEN MADE AVAILABLE?

If the operator replies YES then TTTP will ask for tapes again but, if the operator replies NO, TTTP will output the warning message:

IF YOU GO ON INCREMENTS WILL BE LOST

followed by the question:

OK TO GO ON?

If the operator again replies NO, TTTP will terminate with a

TTTP TERMINATED

message, without having processed the increment. If he replies YES, the increment will be marked as obsolete and the TTTP will go on to the next increment to be processed.

Output tapes

It is possible to specify which tapes TTTP is to try to use as output tapes by including the ONTO parameter in the macro call (see TTTP in Chapter 13 of the manual *GEORGE 3 & 4 Operation Management*).

If it is desired that TTTP should continue to use as output tapes those which it last processed onto, the parameter ONTO OLD should be included, as this is the only way in which this can be done. However, it should be noted that this is only possible if increments are processed contiguously (that is, if processing is to begin with the increment next after the last one processed in the previous run). If no "old" tapes are available TTTP will output the following warning message:

NO OLD OUTPUT TAPES ONTO WHICH TO RESUME

It is possible to request TTTP to begin processing onto specific tapes in :DUMPER's directory, by means of the ONTO *tsn* parameter. This may be repeated for up to six tapes but it should be noted that the tapes should have the name GEO3PROCTAPE and should have been in :DUMPER's directory before LISTDUMP was run.

The algorithm which TTTP uses for getting output tapes (assuming that it is not continuing on "old" tapes) is that it first looks for any tapes which were specified in the macro call. If none was specified, or if not enough were specified, it will next look for any other tapes in :DUMPER's directory called GEO3PROCTAPE which are not currently in

use in :SYSTEM.INCINDEX and will try to use these. Finally, if it still does not have the required number of tapes, it will ask for pool tapes by outputting the message:

TTTP NEEDS n POOL TAPE(S)

where n is the number of tapes required. If these tapes are procured, they are renamed GEO3PROCTAPE.

If TTTP is not allowed pool tapes (that is, if the operator has typed CANTDO LOAD in reply to all requests for pool tapes) it will, if possible, proceed with the number of tapes it has already. If this is not possible, TTTP will output the explanatory message:

NO OUTPUT TAPES AVAILABLE FOR PROCESSING

and will terminate the run,

Updating INCINDEX

INCINDEX is the system file that holds all information relating to increment numbers and the tapes on which they are stored. When an increment is first dumped, a record is added to INCINDEX containing the number of tapes on which that increment is dumped, and their serial numbers. When this increment is processed by TTTP, the tape serial numbers of the newly produced tapes are inserted in the record in front of the existing serial numbers, and the number of tapes is updated accordingly. A single increment may be processed several times by TTTP, and each time the new tape serial numbers are inserted. The TTTP parameter DEPTH determines the number of sets of copies to be kept, (see Chapter 13 of the manual *GEORGE 3 & 4 Operation Management*).

End conditions

Successive increments are processed until one of the following occurs:

- 1 An exception condition such as:
 - (a) End of input tape
 - (b) End of output tape
 - (c) Tape failure etc.
- 2 TTTP reaches the final increment to be processed.

In case 1, there are two possibilities. Recoverable conditions such as end of tape cause a suitable message to be output on the operator's console, followed by the message:

OK TO GO ON?

to which the operator can answer YES or NO. TTTP has built-in error recovery procedures for tape failures, etc., and action is only required of the operators if it actually asks a question at the console; thus displays at the console are purely to give information.

If an irrecoverable error occurs, TTTP will terminate with a

TTTP TERMINATED

message.

If TTTP has found tapes which it can RETURN or REUSE, it will sit waiting for a DUMP to be taken before issuing the commands to release the tapes. The message:

TTTP IS AWAITING A DUMP

is output informing the operators that TTTP is in this state.

In case 2, TTTP will terminate with the message:

TTTP – PROCESSING COMPLETED

If TTTP finishes with this message then tapes that have become obsolete during the run will have been returned automatically. In all other cases, it is left to the discretion of the operator to decide later whether tapes made obsolete by TTTP should be released. If it is decided that they should be, this may be done by the operator issuing a DUMP RETURN or DUMP REUSE command (see Chapter 8).

A report on each run is made by TTTP to the line printer, containing details of which files were copied, which tapes were used, etc. It should be noted that whenever the size of an increment or file is mentioned, the units are file-store blocks (512 words).

At the successful termination of any TTTP run the message:

****** END OF TTTP RUN ******

will be displayed.

Full details about the structure and running of the TTTP program are given in Chapter 4 of the manual *GEORGE 3 & 4 Operation Management*.

MESSAGES

These are categorised as to when they occur.

- 1 Initialisation and parameter analysis
 - (a) TTTPROCESSING UNDER OPERATOR CONTROL
 - (b) TTTPROCESSING UNDER MOP CONTROL
 - (c) ERROR IN PARAMETER *xxxxx*
 - (d) NO OUTPUT TAPES REQUIRED BY TTTP
 - (e) WHICH INCREMENT TO BE PROCESSED FIRST?
 - (f) WHICH INCREMENT TO STOP PROCESSING?
 - (g) SIZE SO FAR: OLD *n*; NEW *m*
 - (h) SIZE NEXT TAPE: OLD *n*; NEW *m*
 - (i) RATIO NOT MET FOR PROCESSING SO FAR
 - (j) RATIO NOT MET FOR NEXT INPUT TAPE
 - (k) LISTDUMP MUST BE RUN IN ORDER TO PROCEED
- 2 Input tapes
 - (a) NEW INPUT TAPE NEEDED FOR PROCESSING
 - (b) NO INPUT TAPES AVAILABLE FOR INCREMENT
 - (c) THE LIST OF POSSIBLE TAPES IS AS FOLLOWS
tsn(W) tsn(W) tsn(W) ... (the (W) is optional and indicates that the tape has been WRONGed)
 - (d) HAVE ANY TAPES NOW BEEN MADE AVAILABLE?
 - (e) IF YOU GO ON INCREMENTS WILL BE LOST
 - (f) NAMELY *n* TO *m* INCLUSIVE
 - (g) READ FAIL ON TAPE *tsn*
- 3 Output tapes
 - (a) NO OLD OUTPUT TAPES ONTO WHICH TO RESUME
 - (b) *tsn* SPECIFIED BUT NOT AVAILABLE
 - (c) TTTP NEEDS *n* POOL TAPE(S)
 - (d) TTTPROCESSING WITH ONLY *n* COPIES
 - (e) I HAVE ONLY *n* OUTPUT TAPES
 - (f) TOO MANY OUTPUT TAPES; LIST TRUNCATED
 - (g) 'COPIES' INCONSISTENT WITH 'ONTO OLD'
 - (h) *tsn* IS STILL LOADED
 - (i) SHALL I UNLOAD IT ALSO?
 - (j) WRITE FAIL ON TAPE *tsn*
 - (k) AN OUTPUT TAPE IS FULL
 - (l) END OF TAPE *tsn* DETECTED
 - (m) HEADER READ FAIL ON A POOL TAPE
- 4 General
 - (a) PLEASE ANSWER YES OR NO
 - (b) OK TO GO ON? (following another message)

- (c) NO MORE CORE AVAILABLE—REALTIME ON?
- (d) TAPE ERROR (SEE ABOVE) (following a transcription error)
- (e) INCREMENT(S) PROCESSED : n TO m
- (f) TTTP FAILED TO POSITION TAPE tsn

5 Premature termination

Messages (a) to (d) are followed by message (e)

- (a) "TO" LESS THAN "FROM" IS NOT ALLOWED
- (b) NO OUTPUT TAPES AVAILABLE FOR PROCESSING
- (c) TTTP IN QUIET MODE SO FORCED TO STOP
- (d) NO MORE CORE AVAILABLE—REALTIME ON?
- (e) TTTP TERMINATED
- (f) TTTP TERMINATED—INCORRECT PARAMETERS
- (g) TTTP TERMINATED—ERRORS DETECTED
- (h) TTTP TERMINATED—INCINDEX CORRUPTED?
- (i) TTTP TERMINATED—COMMAND ERROR
- (j) IS A TTTP POSTMORTEM REQUIRED?
- (k) INCINDEX HAS BEEN RECONSTITUTED
- (l) "FROM" IS n ; "TO" IS m (followed by message (a) above)

Messages (f) to (i) indicate that something has gone wrong. If a rerun is not successful, a bug report should be submitted, accompanied by as much relevant information as possible. A TTTP postmortem should normally be supplied in such cases.

6 Successful termination

- (a) REVISED TTTP SAVED PERMANENTLY (after run in Revision mode)
- (b) TTTP—PROCESSING COMPLETED
- (c) NO TAPES RELEASED
- (d) TTTP IS AWAITING A DUMP
- (e) **** END OF TTTP RUN ****

7 Error numbers

Throughout the running of TTTP, there may occur the message:

ERROR NUMBER n DETECTED

If n is less than 800, TTTP will immediately halt with the message 5g above. A value of n which is greater than or equal to 800, implies either an error while transcribing, or an error while updating INCINDEX. In either case, some form of recovery action is made before TTTP halts with message 5g or 5h. The meanings of the error numbers are given below.

<i>Number</i>	<i>Meaning</i>
400	Failure while trying to read LITTLE INDEX.
500	After transcription error; lost track of the failing input tape.
595	Required tape has not been found.
600	The file INC LENGTHS has been found to be empty.
800	Failure while trying to read INCINDEX.
801	The correct number of tapes to be marked as obsolete have not been found.
802	Expected "dummy" increment in list has not been found.

<i>Number</i>	<i>Meaning</i>
803	Read past expected next increment in INCINDEX.
900	A start sentinel was expected but has not been found.
901	A start of increment sentinel was expected but has not been found.
902	A sentinel was expected but has not been found.
903	An end sentinel was expected but has not been found.
904	The input tape has been found to be past the correct increment.
905	An end sentinel has an incorrect format.
906	End of file found but not expected.
907	Incorrect sentinel count found at end of increment.
910	A long block has been read.
912	A qualifier block was expected but a tape mark has been found.
913	Wrong number of blocks found.
914	The start and end sentinel block counts disagree.
915	Wrong number of sentinels found.
916	Incorrect sentinel count found at end of file.
917	The tape is past the next file which is to be copied.

Chapter 5 Scheduling

The two GEORGE routines that control scheduling are the high level scheduler and the low level scheduler. At any given time, whilst GEORGE is running, there may be a number of jobs loaded and waiting to be run. The function of selecting which of the waiting jobs to run is performed not by the operator but by the high level scheduler. This routine makes its selection on the basis of the urgency, peripheral needs, and so on, of the jobs waiting in the file-store, and the scheduling requirements that have been presented to it by users. The jobs that are selected by the high level scheduler are passed on to the low level scheduler. This routine shares out central processor time in accordance with the policies presented to it by the high level scheduler and, at the same time, ensures reasonable response times to MOP users.

Further information on scheduling and the high and low level scheduler is given in Chapter 8 of the manual *Operating Systems GEORGE 3 and 4*.

THE OPERATION OF THE HIGH LEVEL SCHEDULER

The job description to control the running of the high level scheduler must be contained in the file :SYSTEM.HLSFILE. The scheduling subject program to be used at the installation must be loaded in :SYSTEM.JOBSTARTER and :SYSTEM.HLSEMERGENCY.

If the program loaded from :SYSTEM.JOBSTARTER fails, it is post-mortemed to the monitoring file system and the high level scheduler job initiates another job to schedule the system. The original job then ENDJOBS to allow the post-mortem to be examined.

The new job loads a reserve scheduling program from the file :SYSTEM.HLSEMERGENCY.

At intervals the currently running scheduling job initiates another job to schedule the system and then ENDJOBS to allow the monitoring file to be printed. In this case whichever of :SYSTEM.JOBSTARTER or :SYSTEM.HLSEMERGENCY was being used will be used again. When the scheduling jobs are about to switch, a display will be sent to the operator's console. If there has been a failure, this display will be:

HLS EMERGENCY

If the switching is merely a routine job switch, the display will be:

HLS SWITCHING

If, on trying to load either :SYSTEM.JOBSTARTER or :SYSTEM.HLSEMERGENCY, there is a loading fail, the appropriate message will be sent to the operator's console in the form of a question:

NO :SYSTEM.JOBSTARTER—CAN YOU PROVIDE?

or

NO SYSTEM.HLSEMERGENCY—CAN YOU PROVIDE?

An answer of YES (or Y) will cause the job to try again. An answer of NO (or N) will cause one of the following messages to be output:

SHALL I USE :SYSTEM.HLSEMERGENCY?

or

SHALL I USE :SYSTEM.JOBSTARTER?

An answer of YES (or Y) will cause the job to try this alternative. An answer of NO (or N) is taken to mean that a mistake has been made, and the job will start again, trying to load the original program.

If the job tries the alternative program, and that too is missing, one of the following messages will be output:

NO HLSEMERGENCY EITHER.STARTING AGAIN.

or

NO JOBSTARTER EITHER STARTING AGAIN.

The job then assumes a mistake has been made and starts again from the beginning.

THE COMPUTING POWER INDEX (C.P.I.)

Using the data at its disposal the high level scheduler decides which jobs to run (in practice, which jobs to tell the low level scheduler about). Jobs are initially tentatively started and the high level scheduler must make them fully started before they are allowed to run programs. Typically, most if not all the MOP jobs will be allowed to run programs at the same time, but not all the background jobs. To prevent the system being overloaded with MOP jobs, an upper limit is set on the number of MOP users who can run programs on-line from a MOP terminal.

After deciding which jobs will be made fully started the next step for the high level scheduler is to establish a *Computing Power Index* (C.P.I.) for each one. The C.P.I. for a given job is a fraction representing the proportion of the computer's time to be given to that job.

To illustrate the meaning of the C.P.I., consider the case of three jobs to be run with urgencies F, K and Q respectively. The high level scheduler might assign C.P.I.s which informed the low level scheduler that the F job was to run for 48 minutes in every hour, the K job for 10 minutes and the Q job for 2 minutes. This does not mean that the F job, if it was long enough, could run for 48 minutes continuously, because the C.P.I. is used to determine the proportion of computing time given to a job, not the length of time for which a job can run without interruption. This latter factor is controlled by the low level scheduler.

A particular job's C.P.I. is not absolute, but is relative to the urgencies of the other jobs that are to be time-shared. As jobs start, finish or otherwise change their scheduling requirements, the C.P.I.s of all jobs known to the high level scheduler are adjusted accordingly. To achieve this dynamic control the high level scheduler is activated whenever any of the following occurs:

- 1 The time specified in the installation parameter HLSREVIEW has elapsed.
- 2 A tentatively started job encounters a command which requires it to be fully started.
- 3 A job has ended or been abandoned.
- 4 A job has been broken in on while waiting to be fully started.
- 5 A job has been CONNECTed or DISCONNECTed.
- 6 An EXPRESS command has been issued (see page 58.1).

Thus the high level scheduler is activated whenever scheduling action is necessitated and C.P.I.s might need to be changed.

SAVING BACKGROUND JOBS

When the high level scheduler has decided which jobs to make fully started, and has assigned the relevant C.P.I.s, an attempt is made to save the background jobs that are not going to be made fully started in the near future (if the setting of the switch word allows saving). If the job is saveable it need not remain in core, and will be saved until the job is made fully started, when it will be brought into core. The scheduling of a job will in no way be affected by its being saved.

The reasons for saving jobs are to reduce the amount of core store occupied and to reduce the length of chains that are frequently searched by internal GEORGE routines. It is also possible for jobs that have been saved to be restarted after a system break.

Background jobs are saved for the following reasons:

- 1 The BACKGROUND limit has been reached.
- 2 The job temporarily cannot have all the peripherals which it has NEEDED.
- 3 The job has specified an earliest start time which is more than 5 minutes ahead of the time when the HLS was last "woken up".

Note that jobs with core images and jobs with workfiles, or with vital system files open are not saveable.

URGENCY LEVELS

In order to distinguish jobs that have a high priority from others, there are 26 different urgency levels at which jobs can be run. Urgency A is the highest urgency level, and urgency Z the lowest. Users are allocated time budgets at different urgency levels by the installation manager; time budgets can be given for any urgency levels between A and Y inclusive, and the manager decides which levels are to be used on the installation by deciding at which levels users are to be allocated time. A user must have time available at a particular urgency to run a job at that urgency; if he does not, GEORGE will run the job at the next urgency lower than that specified at which he does have time available. If the user has no time at any other level then the job will be run at urgency Z, for which level time budgets are not checked.

If a job's urgency level is not specified, it will be taken as being of urgency M.

In the initial filestore the directory :OPERATORS has an allowance of 1000 units of time at urgency A. The manager will give this directory sufficient time budgets to allow jobs to be run under it at the required urgency levels.

THE WHATJOBS AND WHATSTATE COMMANDS

The urgency, C.P.I. and saved status of current jobs can be ascertained by use of the WHATJOBS command. The command:

WHATJOBS URG

will cause the urgency of the job to be output on the operator's console under the heading UR, and the C.P.I., as a percentage, under the heading CP. If the job has been saved by the high level scheduler, SAVED will be output in the UR field.

This information is also output after using the WHATSTATE command with the appropriate parameters.

THE EXPRESS COMMAND

The EXPRESS command (see Chapter 8) can be used to inform the high level scheduler that a particular user job is of great urgency and should not be made to wait, when it requires to be made fully started.

Operator communication with the standard subject program is by issuing an EXPRESS command with an extra parameter. The variations of the command which are used by the standard high level scheduler are:

1	EP <i>n</i> ,SW	Make HLS switch as usual
2	EP <i>n</i> ,POST	Take HLS postmortem and change subject programs
3	EP <i>n</i> ,SVON	Allow job saving
4	EP <i>n</i> ,SVOFF	Inhibit job saving
5	EP <i>n</i> ,CDON	Allow budget checking
6	EP <i>n</i> ,CDOFF	Inhibit budget checking
7	EP <i>n</i> ,CCEP	Cancel EP for job <i>n</i>
8	EP <i>n</i> ,EP or EP <i>n</i>	EXPRESS job <i>n</i>

In the above forms of the command, *n* is either null or is the *job number* or *jobname*, *username* of any non-system issued job. If *n* is present the job must be in the system (although commands 1 to 6 are not concerned with expressing the job).

It should be noted that these commands are special forms of the general

EP *n*,*text*

The standard subject program, however, disregards all forms of *text* except those mentioned above, and so ignores any command containing a different character string (that is, it will not then express job *n*).

Chapter 5.1 Accounting

LOG ANALYSIS

In order to charge the user for a job the installation manager needs certain information which is obtained by analysing the records kept by the system. This procedure is known as *log analysis*. One of several methods may be chosen (see *The log analysis system* in Chapter 10 of the manual *GEORGE 3 and 4 Operation Management*). The manager will tell the operator which method is being employed.

The following methods are provided by GEORGE:

- 1 BUILT-IN LOG ANALYSIS This system runs at the end of every job and obtains its information from the job's own monitoring file. It does not affect the operator.
- 2 THE JOURNAL ACCOUNTING PROGRAM This is a GEORGE subject program, #XK7S, which runs periodically, at intervals determined by the installation manager, and obtains its information from a system file known as the System Journal.
- 3 BUILT-OUT LOG ANALYSIS This consists of the GEORGE subject program #XK7L. This is shortly to be withdrawn, and no further information about it is provided in this manual.

Alternatively the manager may write his own program to analyse accounting information in the System Journal.

The Journal Accounting Program

If the Journal Accounting Program #XK7S is in use any of the following messages may be output to the operator's console. Operator action should be as determined by the installation manager.

ILLEGAL JAP JOBNAME/USERNAME
ERROR(S) DETECTED—RELOADING IN 10 MINS
INCREASE COREOBJECT FOR JAP
OK ONE SHOT FINISHED
JAP SWITCHING
WAITING BEFORE DICTIONARY UPDATE
JAP TERMINATED—COMMAND ERROR
JAP TERMINATED—ERROR(S) DETECTED
SJFILE GEN NO LONGER EXISTS
JA JOB ALREADY EXISTS

Chapter 6 System breakdowns and reloading

There are two forms of breakdowns. They are caused by:

- 1 System errors.
- 2 Hardware, other software, operator's, or system limitation errors.

SYSTEM ERRORS

GEORGE first error

When a first error occurs, control will return to Executive which will immediately output on the operator's console:

SYSTEM ERROR *n*

where *n* is any letter from A to Z (I to O being exclusive to GEORGE 4). This letter identifies the type of error that has occurred. Executive will then cause GEORGE to enter its postmortem routine and information on the last GEORGE address processed will be output on the operator's console in the following form:

BROKE IN *segment name* AT ADDRESS *number* OBEYING *instruction*

The postmortem routine will then address all operable 7020 remote data terminals and MOP terminals known to the system and attempt to bring them to a 'clean' state. In the case of a 7020 this may involve continuing the operation currently in progress, for example reading one or two more cards or printing a line of output. Any message being input on an operator's console attached to a 7020 at the time of the break will have the message CANCEL appended to it, and the input will be discarded. (If the operator sees CANCEL being appended to an input message he should refrain from pressing the INPUT button until the system has had an opportunity to indicate the reason for cancelling this message.)

When each MOP or 7020 terminal has reached a quiescent state the message:

THE SYSTEM HAS TEMPORARILY CLOSED DOWN

will be sent to that terminal. In the case of 7020 terminal this message will be sent to the 7020 operator's console or, if this is unavailable, to a line printer attached to the 7020 (if any).

The postmortem routine will output on a central line printer a list of the jobs in core at the time of the first error and this list will indicate the job currently being processed. The postmortem routine may however wait for the MOP and 7020 terminals to be 'cleared' before commencing this listing. The maximum duration of this wait will be two minutes (unless this value was reset at GEORGE restore time by using the CRUNCHTIME GIN macro) and any attempts to clear MOP or 7020 terminals will be abandoned if they take longer than this amount of time.

The following message is then output on the console:

POSTMORTEM REQUIRED ?

The operator can reply to this message in one of the following ways:

N or NO	Results in suppression of postmortem output
L or LIST	Results in postmortem listing on line printer
D or DUMP	Results in postmortem summary listing on line printer (if available), and a dump to magnetic tape
LD	Results in postmortem listing on line printer followed by postmortem dump to magnetic tape
MD	Results in mini-postmortem listing on a line printer followed by a postmortem dump to magnetic tape

RIn Causes postmortem to right the line printer with geographical number *n*

LINE PRINTER OUTPUT

When a listing is requested (**LIST**, **LD** or **MD** directives) the output line printer unit can be specified by appending the geographical number of the chosen unit to the directive: for example **LIST, 14** would ensure that output is to unit number 14. If the geographical number is omitted, output is to the lowest numbered line printer unit available.

If the line printer unit is in a **WRONGed** state when the first error occurs, the unit can be made operable by using the **RIn** message (where *n* is the geographical number of the line printer unit). This message can only be input after the **POSTMORTEM REQUIRED?** message has been output and before an acceptable postmortem directive has been input.

If all line printer units are in a **WRONGed** state when the first error occurs, the list of jobs in core is output to the operator's console instead of to a line printer. This list will then be preceded by the message

NO LP AVAILABLE FOR POSTMORTEM

If, after the **POSTMORTEM REQUIRED?** message is output, a line printer unit is made operable, or if a line printer unit is specified for output other than the lowest available when the first error occurred, the list of jobs in core is repeated preceding the postmortem listing.

POSTMORTEM DUMP TO MAGNETIC TAPE

The postmortem routine will only dump to magnetic tapes labelled either:

GEORGEPMTAPE

or

SCRATCHVTAPE

If the tape is labelled **GEORGEPMTAPE**, it will first be positioned before the trailer label, unless otherwise indicated by console input messages (see below) and then the dump will be given. If it is labelled **SCRATCHVTAPE**, the tape will be renamed **GEORGEPMTAPE** and the dump will be written immediately after the header label.

When a dump to magnetic tape is required, the following will be output on the console:

DUMP TAPE TSN, DECK NO?

The postmortem routine then loops awaiting the reply, the format of which is:

tape serial number, deck number

Note that there is no space between the comma and the second parameter. For the *tape serial number*, up to eight octal, numeric characters will be accepted; *deck number* is the geographical number of the tape deck on which the tape has been or will be loaded. If the reply is invalid, the question '**DUMP TAPE TSN, DECK NO.?**' will be repeated. If the reply is valid, the message:

START OF DUMP NUMBER *n*

will be output immediately before the dump starts, where *n* is the number of dumps on the tape when the present dump has finished.

CHANGING THE POSTMORTEM DUMP TAPE

It is possible to change the dump tape, so long as the dump has not started, after the following two messages:

LOAD TAPE *tape serial number*

TAPE *tape serial number* LOAD WPR

This is done by typing:

USE TAPE *tape serial number, deck number*

or

UT *tape serial number, deck number*

Where *tape serial number* is the tape serial number of the magnetic tape to be used and *deck number* is the tape deck on which the tape is mounted.

The dump tape may also be changed if, for any reason, the original request **DUMP TAPE TSN, DECK NO.?** is

repeated, whereafter the input should be the normal reply as described above. In each of the input messages the *tape serial number* and *deck number* can be followed by the character string BOT, for example:

tape serial number,deck number,BOT

USE TAPE *tape serial number,deck number,BOT*

UT *tape serial number,deck number,BOT*

This character string will cause the postmortem dump to be written to the tape immediately after the header label and, in effect, erases all previous postmortem dumps written to the tape.

While dumping is being attempted, the following messages may be output:

- 1 INVALID MESSAGE The 'USE TAPE' input message is invalid, for example the *tape serial number* is not an octal number. Postmortem will continue to wait for the original tape to be loaded.
- 2 TAPE *tape serial number* LOAD WPR
- 3 LOAD TAPE *tape serial number*
- 4 TAPE ON DECK *n* NON-STANDARD HEADER LABEL An error has been found in the header label of the postmortem dump tape. The tape will be rewound and postmortem repeats the request 'DUMP TAPE TSN, DECK NO.?'
- 5 TAPE *tape serial number* INCORRECT HEADER LABEL The tape requested is not correctly labelled. Postmortem repeats the request 'DUMP TAPE TSN, DECK NO.?'
- 6 *tape serial number* IS NOT ON DECK *n* The tape specified in the input is not the tape held on the deck number given.
- 7 UNIT *n* FAIL *m* There has been a tape fail while dumping. Postmortem will rewind the tape and repeat the request 'DUMP TAPE TSN, DECK NO.?' so that the dump can be restarted on another tape.

Before the postmortem dump starts the message 'START OF DUMP NUMBER *n*' is output on the operator's console. *n* signifies the position of the dump on the tape (*n* = 1 for the first dump).

At the end of the postmortem dump to tape the following message is output:

END OF PM DUMP

ACTION AFTER POSTMORTEM COMPLETE

When the specified action has been completed, postmortem checks on the state of all peripherals before continuing.

Enough time is allowed for all peripheral transfers to terminate. If a peripheral is found to be in a 'busy' state after this time, the following message is output to the operator's console:

UNIT *n* BUSY-OPAT

where a magnetic tape unit is busy. It is assumed that the deck is rewinding and three minutes are allowed for this transfer to terminate. The operator is advised of this by the message:

AWAITING REWIND:DECK *n*

If after the allowed time the tape deck is still in a 'busy' state, the message

UNIT *n* BUSY-OPAT

is then output.

Finally, depending on the value of the installation parameter RELOAD, either the same version of GEORGE is automatically reloaded or the operator is invited to specify another version (see page 7).

LISTING POSTMORTEM DUMPS FROM MAGNETIC TAPE

A postmortem dump on magnetic tape can later be listed either under GEORGE using the PMPRINT macro (see Chapter 8) or under an operator's Executive using the #XK65 program.

The #XK65 program can be used to list all or specified postmortems held on a GEORGE PMTAPE tape. The hardware requirements for running the program are:

2K words of core store (approximately)

1 line printer
1 magnetic tape unit

When the program is loaded it will HALT. The operator then has three options:

- 1 He can print all the postmortems held on a GEORGE PMTAPE tape, by typing:

GO #XK65 20

- 2 He can print the identities of all the postmortems held on a GEORGE PMTAPE tape, by typing:

GO #XK65 21

- 3 He can print selected postmortems by typing:

ON #XK65 *number*₁, *number*₂ . . . *number*_{*n*}

where each *number* specifies the position of one postmortem in a series of postmortems held on a GEORGE PMTAPE.

The program is then entered by typing:

GO #XK65 22

Exception conditions

During a run of #XK65 any of the following messages may be displayed by the program on the operator's console:

Message	Meaning	Action
WORD 30 INCORRECTLY SET - RESET & GO	Bit 0 of word 30 is set or word 30 is zero	Set word 30 to a meaningful value and restart the program.
MAGNETIC TAPE GEORGE PMTAPE - FAIL, DELETED MT	A failure has been detected during a magnetic tape PERI.	Abandon the run.
MAGNETIC TAPE GEORGE PMTAPE FORMAT ERROR, DELETED ER	Format error found on postmortem tape (for example extra tape mark found)	Abandon the run.
NO POSTMORTEM DUMPS ON MT <i>tape serial number</i> , DELETED OK	As message	Locate correct tape and re-run.
DUMP <i>nn</i> NOT FOUND, DELETED OK	There are less than <i>nn</i> dumps on the tape	Specify correct dump and re-run.
PAPER LOW	As message	Reset the paper and restart the program.
HALT LP	The program requires a line printer	Make a line printer available and restart the program.

GEORGE second error

A GEORGE second error is an error which occurs at any stage between the occurrence of a GEORGE first error and the loading of GELL. When there is a GEORGE second error, the operator receives the message:

SYSTEM ERROR *n*

where *n* is again a letter identifying the type of error that has occurred. Executive then re-enters postmortem which goes into an immediate loop-stop. An Executive postmortem should be taken at this point. After this the only way of continuing is by reloading Executive and following the normal loading procedure.

Loading after system error

The action taken by GEORGE at the end of a postmortem depends upon the value of the installation parameter RELOAD. This can be set to either:

0 which means that GEORGE is to reload itself

or

1 which means that the operator should be asked what to load.

If RELOAD is set to 0, the same version of GEORGE will be reloaded without any operator intervention being required.

If RELOAD has a value of 1, Executive's GELL will be entered and the following message output on the operator's console:

READY

This gives the operator the option of loading another version of GEORGE. Once he has specified the version to be loaded, the procedure will be the same as for the initial loading of GEORGE (see Chapter 2, page 11).

In the above two cases, Executive is automatically reloaded before GEORGE.

HARDWARE, OTHER SOFTWARE, OPERATOR OR SYSTEM LIMITATION ERRORS

When the first error is caused by hardware, other software, operators or general limitations of the system, post-mortem is entered but subsequent action differs from the normal first error action,

One of the following messages, indicating the nature of the fault, is output on the operator's console:

PUT LOGICAL FILE n ONLINE OR ASK FOR A GENERAL RESTORE

RESIDENCE 3 DOES NOT CONTAIN A FILESTORE

FILESTORE MUST BE MARK $n +$ or $- 1$

NEED LOGICAL FILE 3: CORRECT FSDESC OR PUT FILE ONLINE

NO FILESTORE DESCRIPTION

FILESTORE RESIDENCES FULL

MAGNETIC TAPE DECK BUSY

followed by

POSTMORTEM ENTERED

The RELOAD installation parameter is then automatically set to 1 and the action is as described under *Loading after system error*, page 62).

No postmortem listing is produced and no action is required of the operator before the GELL READY message is output.

OTHER SYSTEM BREAKDOWNS

There may occasionally be a system breakdown without a GEORGE first error occurring. If this happens, the operator may deliberately enter the GEORGE postmortem routine in the following way:

- 1 Press the F5 button (or Control and the appropriate other key on a 1903A, 1902S or 1903S) on his console so that he can communicate directly with Executive rather than with GEORGE.
- 2 Type the message:

POSTMORTEM

This message can be abbreviated, but the first four characters at least must be typed to ensure correct recognition of the message. The postmortem action is then the same as for the first error entry.

Chapter 7 Closing down the system

OBJECT

- 1 To clear all background jobs out of the system.
- 2 To get all MOP users out of the system and to ensure that all their monitoring files are produced.
- 3 To produce a final incremental dump at the end of the session.

METHOD OF CLOSEDOWN

The objectives listed above are realized by means of the FINISH command. The FINISH command may have one of three formats and the consequent actions for each are as follows:

1 FINISH

This is the least urgent form of the FINISH command. The high level scheduler will be informed and no more jobs will be made fully started. Tentatively started jobs will be run until they have finished or require to be made fully started. Fully started background jobs will be allowed to continue until they have finished. LISTFILES waiting to be output can be initiated.

When all background jobs have finished, the message

ALL BACKGROUND JOBS FINISHED : *n* MOPS JOBS LEFT

is output to the operator's console. When all jobs, LISTFILES and INPUTs have finished, the message

ALL USER JOBS NOW FINISHED

is output.

2 FINISH *length of time*

The high level scheduler will be informed and no more jobs will be made fully started. Tentatively started jobs will be run until they have finished or require to be made fully started. Fully started background jobs will be allowed to continue until the *length of time* given as a parameter to the command has elapsed, when FINISH NOW action (see below) will be taken for them. Background jobs and MOP jobs in limbo (see *Limbo*, page 42.5) may be connected until the *length of time* specified expires. LISTFILES waiting to be output can be initiated provided that this takes place before the time given has passed. At the end of this time any LISTFILES and INPUTs in the above categories will be allowed to finish. LISTFILES to the monitoring file and INPUTs in USER context will be abandoned when the job issuing them is abandoned.

When all background jobs have finished, provided it is before the time in the command parameter, the message

ALL BACKGROUND JOBS FINISHED : *n* MOP JOBS LEFT

is output on the operator's console. When all jobs, LISTFILES and INPUTs have finished, the message

ALL USER JOBS NOW FINISHED

is output.

The length of time may be extended or shortened by issuing another FINISH command provided that this occurs before FINISH NOW action has been taken.

3 FINISH NOW

This is the most urgent form of the FINISH command. The high level scheduler will be informed so that it may take appropriate action and no more jobs will be accepted by the system or be made fully started. Any jobs already in the system will be run until they are finished or may be abandoned and all LISTFILES and

INPUTs in progress will be abandoned. When all jobs (including those issued by the system) have finished or been abandoned, the message

ALL USER JOBS NOW FINISHED

is output on the operator's console

The operator is then asked if a dump is required. If it is, the DUMP command should be used. When the dump is complete, and its monitoring file has been output, the operator should type

LX

TAPES

This will give a printout of the DUMPER history giving details of old dump tapes and dates on which they were produced, and a list of all the tapes known to the system.

When all activities have been terminated, the message

EVERYTHING HAS FINISHED

is output on the operator's console and the system can now be closed down. The operator should press F5 (or Control and the appropriate other key on a 1903A, 1902S or 1903S) and input DELETE followed by ACCEPT. The system will then be deleted.

Cancellation of FINISH

If the CANCEL FINISH command is issued, provided that FINISH NOW action has not already been initiated, the ban on accepting jobs will be lifted and they may be introduced subject to the usual JOBLIMIT restrictions.

Any jobs already abandoned must be restarted by the operators. Any LISTFILE or INPUT commands not permitted after FINISH must be reissued.

Important note

Do not press F5 and input DELETE or POSTMORTEM when there are jobs running as this will leave users' files in an indeterminate state.

Chapter 8 Specifications of GEORGE commands

INTRODUCTION

This chapter lists in alphabetical order with their specifications those GEORGE commands that come into one of the following categories:

- 1 Commands that may be issued from the operator's console.
- 2 Commands which may not be issued from the operator's console but about which the operator may on occasion need to know, for example, to correct an error on the card containing the command. Commands currently in this category are DISENGAGE, DOCUMENT, INPUT and JOB.

The specification of each command is divided into sections as described below.

The operator should always press the INPUT button on his console before typing a command.

Name

Each specification is headed by the name of the command (that is, the verb) followed, in brackets, by an abbreviation (usually of two letters) if any abbreviation is permitted. If an abbreviation is given in the heading, it may always be used in place of the full command verb.

Function

This is a single sentence description of the use of the command.

Format

A description of the format of the command is given under this heading. The following is a summary of the general format of a command:

- 1 A command consists of a verb normally followed by one or more parameters.
- 2 There must be at least one space between the verb and the first parameter.
- 3 Parameters must be separated from each other by commas.
- 4 Spaces before the verb and spaces between parameters are not significant.
- 5 The multiline character — (hyphen), indicating that a command continues on the next line, cannot be issued direct from an operator's console. It will cause the following message to be displayed:

ERROR: THE MULTILINE CHARACTER IS ILLEGAL FROM AN OPERATOR'S CONSOLE

Example

optional spaces VERB *space* PARAMETER *optional spaces*, PARAMETER

PARAMETER FORMATS

The parameters of the commands are given in italics, and notes about the formats of certain parameters which are used with a number of operator commands are given below.

username A *username* consists of a colon followed by up to twelve letters, digits, spaces or hyphens, beginning with a letter. Spaces before the first letter are ignored. For example:

:FRED

:A12345678901

tape serial number or *tsn* A *tape serial number* is an octal integer less than #40000000. It should not be preceded by #, and must be enclosed in parentheses. For example:

(10242)

(3777777)

<i>jobname</i>	A <i>jobname</i> consists of up to twelve letters, digits or hyphens beginning with a letter. For example: JOBNO1 EDITJOB
<i>text</i>	A <i>text</i> may take any form provided that it conforms to the rules that govern all parameters. This means that parentheses, brackets or quotes may be included provided that they are paired but they will be treated as characters of text. A comma may be included only within parentheses, brackets or quotes. Internal spaces within texts are always significant. For example: PROG RUN COMPLETE (*CRO NEEDED)
<i>parameter name and local file name</i>	These have the same format, namely, up to twelve letters, digits, spaces or hyphens, beginning with a letter. Leading spaces are ignored, and trailing spaces have no effect as the name is space-filled up to twelve characters by the system. For example: FRED A12345678901
<i>magnetic tape description</i>	A <i>magnetic tape description</i>, as required in the LABEL command, has the following format: (tape serial number, magnetic tape name (details))(*MT) where <i>magnetic tape name (details)</i> and (*MT) are optional. The format of <i>magnetic tape name (details)</i> is specified in Chapter 10 of the manual <i>Operating Systems GEORGE 3 & 4</i>.
<i>peripheral description</i>	A <i>peripheral description</i> is the character string that uniquely identifies a local or remote peripheral connected to the central processor. Its format depends on the way in which the peripheral is connected to the central processor: - For a local peripheral the format is: <i>n</i> or <i>Un</i> where <i>n</i> is the geographic number of the peripheral - For a remote peripheral connected via a character buffering multiplexer or uniplexer the format is either - FULL FORM: <i>Un.Lm.laax</i> - or - SHORT FORM: <i>laax</i> where <i>n</i> = geographic number of the multiplexer/uniplexer <i>m</i> = line number <i>aa</i> = 7020 terminal identifier; a two-character string of which the first must be a letter <i>x</i> = interface number of the 7020 which is in the range 1 to 6 If <i>x</i> is omitted then the lowest interface number available is used. - For a remote peripheral connected via a message buffering communications processor the format is either - FULL FORM: <i>Un.lzzzz</i> - or - SHORT FORM: <i>lzzzz</i> where <i>n</i> = geographic number of the communications processor <i>zzzz</i> = identifier of the device which must be in the range 1 to 4095
<i>peripheral type</i>	A <i>peripheral type</i> is an asterisk followed by either a standard two character code or the two character decimal equivalent (the leading zero must be included). For example a line printer may be identified by *LP or *02. Spaces are not significant.

A complete list of peripheral types is given below.

<i>Peripheral type</i>	<i>Type number</i>	<i>Description</i>
*TR	00	Paper tape reader
*TP	01	Paper tape punch
*LP	02	Line printer
*CR	03	Card reader
*CP	04	Card punch
*MT	05	Magnetic tape
*DA, *ED	06	Direct access device available under UDAS (see note 1 below)
*DR	09	Magnetic drum (see note 2)
*UX (*IT)	10	Uniplexer
*MX	11	Multiplexer
*FD	13	Fixed disc (see note 1 below)
*ID	14	Alphanumeric visual display control unit
*GP	20	Graph plotter
*GD	21	General purpose visual display unit
*UD	23	Universal document reader
*CC	24	7900 communications processor I.P.B.
*HD	25	High speed magnetic drum (see note 2)
*PB	28	Inter-program communication link
*FR	50	File reader
*FW	51	File writer
Standard types	52 to 59	Various exotics
*CI	60	Command issuer

Notes

- Peripheral type *DA will eventually replace both *ED and *FD.
- Magnetic drums types 09 (*DR) and 25 (*HD) are generally for use by GEORGE only; the programmer is not allowed to on-line them. However the facility does exist for engineers' programs to do so.

property name

A *property name* is a string of up to 12 characters (letters, digits, hyphens and spaces). The character string must commence with a letter and spaces are significant. A property name parameter is introduced by the parameter prefix PROPERTY or PR. Spaces between the parameter prefix and the character string are not significant. Where *property names* is specified, the parameter prefix is followed by one or more property names separated by ampersands (&).

Examples

PR PP160

PR INVOICE&PP160

peripheral property name

A *peripheral property name* is the property of a magnetic tape deck. It is always preceded by the parameter prefix PROPERTY (or PR) and it may only be one of the following:

TRACK7	7 track deck
PE	9 track phase encoded deck
NRZI	9 track NRZI deck
TRACK9	9 track deck
MODE n	deck in recording mode n , where n must be a decimal number of not more than 2 characters.
FAST	fast deck
SLOW	slow deck

Where *peripheral property names* is specified, the parameter prefix is followed by one or more peripheral property names separated by ampersands (&).

Example

PR TRACK7&FAST&MODE36

PARAMETER QUALIFIERS

Some parameters may include qualifiers. Qualifiers may be given in any order, with or without intervening spaces. They must be separated from each other by commas and the whole set enclosed in parentheses. The qualifiers relevant to each command are normally given in the command specification. A complete list appears in Chapter 10 of the manual *Operating Systems GEORGE 3 and 4*. Some qualifiers which may appear in operator commands are given here.

<i>Qualifier</i>	<i>Description</i>								
*MT	This is used only with magnetic tape descriptions. It is obligatory in certain commands.								
PROPERTY <i>magnetic tape properties</i> or PR <i>magnetic tape properties</i>	This qualifier may be used to qualify the magnetic tape descriptions in the NEW command, and each <i>magnetic tape property</i> must be one of the following: <table><tr><td>TRACK7</td><td>7 track tape</td></tr><tr><td>PE</td><td>9 track phase encoded tape</td></tr><tr><td>NRZ I</td><td>9 track NRZI tape</td></tr><tr><td>TRACK9</td><td>9 track tape</td></tr></table> A suitable combination of the above properties may be specified.	TRACK7	7 track tape	PE	9 track phase encoded tape	NRZ I	9 track NRZI tape	TRACK9	9 track tape
TRACK7	7 track tape								
PE	9 track phase encoded tape								
NRZ I	9 track NRZI tape								
TRACK9	9 track tape								
PROPERTY <i>peripheral property names</i> or PR <i>peripheral property names</i>	This qualifier may be used to qualify the peripheral type *MT in the UNLOAD command. The <i>peripheral property names</i> must be properties of magnetic tape decks, as defined under <i>peripheral property name</i> on page 67. Only the properties TRACK7, PE, NRZI and TRACK9 will be effective.								

Forbidden contexts

Under this heading are listed the names of the contexts in which it is forbidden to issue the command. There are several contexts recognised by GEORGE, but the operator need only be familiar with a few of them.

OPERATOR, NOT OPERATOR

If a command is typed on an operator's console, it is said to be issued in OPERATOR context. If, therefore, NOT OPERATOR is given as a forbidden context, the command may only be issued from an operator's console. Most of the commands in this chapter are in this category.

However certain commands specified in this chapter may not be issued from the operator's console (see category 2 in the introduction to the chapter). This fact is indicated by the forbidden context OPERATOR (or by NO USER or NOT READER, see below).

READER, NOT READER

A command which is read by a card or paper tape reader is said to be issued in READER context. If, therefore, NOT READER is given as a forbidden context, the command may only be read from card or paper tape, and may not be typed at the operator's console (for example, the DISENGAGE command).

USER, NO USER

A command issued in USER context is one that is issued from within a job. Therefore a command for which NO USER is a forbidden context may only be issued from within a job, and not from an operator's console. (OPERATOR and READER contexts are both subsets of NO USER context).

REMOTE, NOT REMOTE

A command issued in REMOTE context is one that is issued from devices or operator's consoles connected to the central processor via telephone or telegraph lines to a multiplexer, uniplexer or communications processor.

Execution

This section contains details about the implementation of the command that are relevant to the operator.

Error Messages

Under this heading for each command are listed the command error messages peculiar to that command.

COMMON COMMAND ERRORS

GEORGE will also issue a command error message if the system does not recognise a command, owing to errors in

the format of the verb or the parameters, or if a command is issued in the wrong context. (Errors in commands issued in READER context are reported on the appropriate operator's console; that is, the central or cluster console.) The error messages which may appear on the operator's console in these cases are given below in the appropriate sections. These messages are usually prefixed by the words

ERROR IN *command*

where *command* is the first word of the command (that is, the verb); for example:

ERROR IN LOAD

Errors in verbs

If an error is found in the verb itself, and if the error is such as to make the format of the verb incorrect (that is, not starting with a letter or being more than twelve characters long), then the error message will be prefixed by the words

ERROR IN VERB

This is the case with the first two error messages below.

VERB FORMAT ERROR	First word of the command does not begin with a letter.
VERB TOO LONG	First word of the command is more than 12 characters long.
THIS IS NOT A COMMAND	The command is not known to the system, or a user macro has been issued in NO USER context.

Context errors

If a command is issued in the wrong context (see *Forbidden contexts* above) the following message is output on the operator's console:

THIS COMMAND IS NOT ALLOWED IN *context name* CONTEXT

where *context name* may be any one of the following:

NO USER
NOT READER
OPERATOR
READER
REMOTE

For example, if, by mistake, a card pack with a LOAD command as the first card were put in the card reader, GEORGE would read the first card and then output on the operator's console the error message:

ERROR IN LOAD: THIS COMMAND IS NOT ALLOWED IN NO USER CONTEXT.

Errors in parameters

Any of the following error messages may be output by the system if an error is found in the format of a command parameter or parameters.

z PARAMETER MISSING
z PARAMETER FORMAT ERROR
z PARAMETER NULL
PARAMETER FORMAT ERROR: z
PARAMETER IS LONGER THAN THE PERMITTED MAXIMUM (511 characters)
z IS NOT A VALID PERIPHERAL DESCRIPTION PARAMETER
PERIPHERAL DESCRIPTION PARAMETER MISSING
FORMAT ERROR IN USERNAME
USERNAME IS NOT IN DICTIONARY
USERNAME/LOCAL NAME TOO LONG
USERNAME: 2ND CHARACTER MUST BE A LETTER
FORMAT ERROR IN TSN

ABANDON (AB)

Function

Allows the operator to abandon the execution of a job.

Format

ABANDON *jobname,username*

Forbidden contexts

NOT OPERATOR

Execution

The command causes the specified MOP job, whether or not it is in limbo (see *Limbo*, page 42.5), or background job to be abandoned. All the job's files are closed, and any core image that may exist is deleted, and any peripherals being used by the job are released.

The messages

ABANDONED BY OPERATOR INTERVENTION

and

END OF MACRO

are sent to the monitoring file, the latter being issued as each of the job's files, for example, the job description file, is closed down. In addition the message

jobname, username WILL BE ABANDONED AS SOON AS POSSIBLE

is output on the operator's console if the job cannot be abandoned immediately. This may happen if the job is in the middle of a command that cannot be interrupted such as a LOGOUT or ENDJOB or if the job is being broken-in on. If the job is logging out then the error message

z DOES NOT EXIST

may be output afterwards.

The message

jobname, username IS BEING ABANDONED

is output on the operator's console when the job is found to be in an abandonable state.

To implement the command, GEORGE simulates a LOGOUT (for a MOP job) or an ENDJOB (for a background job). The log analysis program is then entered to process the job's monitoring file. The monitoring file will always be listed in full, but not RETAINED. In order that the charge for running an abandoned job may be modifiable, the ABANDONED message is stored in the LOGGING category. A system issued job is not abandonable. If an attempt is made to abandon a job that does not exist the operator will be informed.

Example

Format	Resulting messages	Explanation
ABANDON JOBNO135,:FRED	1 (to monitoring file system) ABANDONED BY OPERATOR INTERVENTION END OF MACRO	:FRED's job JOBNO135 is abandoned.
	2 (to operator's console) JOBNO135, :FRED IS BEING ABANDONED	

Note

This command must be used with great care. It is intended that it should be used only when a serious jam has arisen and the only alternative would be to reload the whole system.

ABANDON (AB)

Error messages

JOB NAME PARAMETER MISSING

JOB *z* NOT ABANDONABLE : SYSTEM ISSUED

USER NAME PARAMETER MISSING

z DOES NOT EXIST

z IS NOT A CORRECTLY FORMED NAME (job name or user name)

JOB NOT ABANDONABLE : IT IS NOT IN YOUR CLUSTER

Logging messages

ABANDONED BY OPERATOR INTERVENTION

time milltime FINISHED : *number*₁ LISTFILES HERE, *number*₂ ELSEWHERE AND FURTHER LISTFILES
FOR *number*₃ MULTIFILES

Relevant parts of the message will be omitted if *number*₂ or *number*₃ is zero.

ANSWER (AN)

Function

Gives the answer to a QUESTION command.

Format

ANSWER *username, jobname, text*

The *user name* and the *job name* must be those specified in the output on the operator's console of the QUESTION command.

The *text* must not include the device-dependent 'cancel-character' character (usually ^).

Execution

The operator should issue this command when he receives a message of the form:

user name, job name QUESTION: *text*

This message is repeated periodically until an ANSWER command is given.

A check is made that the job specified by the operator in the ANSWER command is waiting in a QUESTION command.

A MOP user may provide the answer to a question output on the operator's console. If he does so before the operator has issued an ANSWER command, the message:

user name, job name ANSWER: *text*

is sent to the operator's console and the operator should not attempt to provide an answer.

If both the operator and the MOP user attempt to answer a question, one of them will receive a command error message indicating that the job has not asked a question (see *Error messages* below).

The operator may issue an ANSWER command after the MOP user has broken in.

Example

Message received at operator's console:

:FRED, EDITJOB QUESTION: IS *CR0 READY?

Answer provided by operator:

ANSWER :FRED, EDITJOB, *CR0 IS READY

If the question above was answered by the MOP user, the following message would be output at the operator's console:

:FRED, EDITJOB ANSWER: *CR0 IS READY

Error messages

JOB HAS NOT ASKED A QUESTION

NO SUCH JOB

ONLY ONE PARAMETER ALLOWED

USER/JOB NAME PARAMETER MISSING

z IS NOT A CORRECTLY FORMED NAME (job name or user-name)

ASSOCIATE (AE)

Function

Set up an association between commands issued by a job initiated from a device and a property known to the installation, and is used to route output from jobs to suitable destinations.

Format

ASSOCIATE *peripheral description*, PR *property name*, *command parameter*

The parameters may be in any order.

The *peripheral description* is as described on page 66.

The *command parameter* takes the form

verb peripheral type peripheral type . . .

where the verb may be expressed in either full or short form and the peripheral type parameters are optional. If present, the parameters qualify the association. If absent, GEORGE takes default values that are equivalent to specifying all suitable peripheral types. See *Examples* below.

Forbidden contexts

Not (central) OPERATOR

Execution

The following checks are made before the association is made:

- 1 The command has the correct format.
- 2 The peripheral description is valid and is known to the installation description file.
- 3 The property name exists.
- 4 The *command parameter* contains a valid *verb* and any *peripheral type* parameters are valid for the *verb* specified.
- 5 The association does not already exist. Normally, if, for the peripheral specified, all or part of the command specified in the command parameter is already associated with a property, a command error will be generated by the ASSOCIATE. However, no error will be generated if the command parameter and property parameter specify an extension to an already existing association. Thus, in the examples below, if a user were to issue the first example followed by the second a command error would result, but the third example followed by the fourth would not cause a command error.

When the ASSOCIATE command has been successfully executed, the following message is output to the operator's console:

ASSOCIATION *z* NOW EXISTS

where *z* is the parameter string of the ASSOCIATE command.

Examples

AE U29.L10.IAA5,PR OXFORD, LF *TP *LP

If a LISTFILE is issued from the remote tape reader which is on interface 5 of a 7020 which has terminal identifier AA and is connected to the main processor via line 10 of multiplexer unit 29, then any tape punch and line printer output should be sent to the tape punch and line printer that have the property OXFORD ATTRIBUTED to them.

AE U29.L10.IAA5,PR OXFORD, LF

If OXFORD is a console property then *TP and *LP will be assumed. If OXFORD is not a console property then *TP, *LP and *CP will be assumed.

ASSOCIATE (AE)

ASSOCIATE U29.I200,PR BRACKNELL, LF *LP

Associates the property BRACKNELL with LF command issued from identifier 200 on a communications processor if the output is for a line printer.

ASSOCIATE U29.I200, PR BRACKNELL, LF

If BRACKNEL is a console property then *TP and *LP will be assumed. If BRACKNELL is not a console property the *TP, *LP and *CP will be assumed.

AE 5,PR A,LF *LP
AE 5,PR B,LF *TP
AE 5,PR C,LF *CP

If a LISTFILE is issued from the central card reader (operator's number 5) then any line printer output should be sent to the line printer that has the property A attributed to it; any tape punch output should be sent to the tape punch with the property B and any card punch output sent to the card punch with the property C. The devices may be either local or remote.

Notes

The following restrictions exist in the current Mark:

- 1 LISTFILE is the only verb allowed in a *command parameter*.
- 2 The maximum number of associations permitted for any one peripheral is 3.
- 3 Normally it will only be useful for input terminal peripherals and CONSOLE properties to be associated with commands, but the command is not restricted to these purposes.

Note that a CONSOLE property is ATTRIBUTED to each peripheral in a cluster. When an ONLINE or LISTFILE command is issued, the CONSOLE property name may be included as a parameter. Output will then be directed to an appropriate peripheral in the cluster sharing the CONSOLE property name. (See page 67 for details of the property name parameter format.)

Error messages

COMMAND PARAMETER MISSING

verb IS NOT A VALID COMMAND PARAMETER

PERIPHERAL *peripheral type* NOT ALLOWED

PROPERTY *property name* UNKNOWN

ASSOCIATION ALREADY DEFINED

ONLY *n* ASSOCIATIONS ALLOWED

PARAMETER MISSING

PARAMETER FORMAT ERROR: *n*

CANCELLATION OF ASSOCIATIONS

Function

Removes an association from the system.

Format

CANCEL ASSOCIATE *peripheral description*, PR *property name*, *command parameter*

where the parameters are as described above.

Forbidden contexts

NOT (local) OPERATOR

Execution

The specified association is removed, if it exists, and the following message is output on the operator's console:

ASSOCIATION *z* NOW NON-EXISTENT

where *z* is the parameter string of the ASSOCIATE command.

Error messages

As for the ASSOCIATE command, *excluding*:

ASSOCIATION ALREADY DEFINED

ONLY *n* ASSOCIATIONS ALLOWED

ATTACH (AH)

Function

Attaches a line of a character buffering multiplexer or uniplexer to a named conceptual, or attaches an identifier of a message buffering multiplexer to a named conceptual.

Format

1 For attachment of lines to a conceptual (for a character buffering system):

(a) ATTACH *unit number.line number,conceptual name.line group*

(b) ATTACH *unit number,conceptual name.line group*

Format (a) is used by the central operator to attach a line of a multiplexer to a given conceptual.

Format (b) is used by the central operator to attach a line of a uniplexer to a given conceptual.

2 For attachment of identifiers to a conceptual (for a message buffering system).

(a) ATTACH *unit number.identifier number,conceptual name.identifier group*

(b) ATTACH *conceptual name.identifier group*

Format (a) is used by the central or remote operator to attach an identifier on a communications processor to a given conceptual.

Format (b) is used by the remote operator to attach his own console to a given conceptual.

Note: Format (a) is only permitted from a remote operator's console for the attaching of identifiers for which it is the nominated console.

The two parameters in formats 1(a) and 2(a) may be in either order.

The *unit number* must be preceded by a U, for example U10.

The *line group* must be expressed in the form *L numerical expression* where the *numerical expression* can take any of the following three forms:

1 <i>n</i>	specifies a single line
2 <i>n(m)</i>	specifies a group of <i>m</i> lines starting with line <i>n</i>
3 <i>(n₁,n₂,n₃ . . .)</i>	specifies a list of line numbers where <i>n₁,n₂,n₃</i> are numbers

The *identifier group* must be expressed in the form *I numerical expression* where the *numerical expression* can take any of the forms described above.

The *line group* or *identifier group* parameters are optional and if they are absent the attachment is made to the next free conceptual line or identifier. If one of these parameters is present it indicates which conceptual line numbers or identifier numbers may be used for attaching the physical line. The first free conceptual line number or identifier number of those specified is the one used.

The ATTACH command may also be issued by a user from a MOP terminal: details are given in Chapter 12 of the manual *Operating Systems GEORGE 3 and 4*.

Execution

In the case of attachment of a character buffering multiplexer line, a check is made to see if a suitable conceptual line is available and, if it is not, a command error is reported. Checks are then made to see that the actual line is not busy, the unit is not being MOPPED OFF and the conceptual does not have any message buffering system identifiers attached. If any of these checks fail, a command error is reported.

The connection is then made and the entry for the specified actual line in the Installation Description File (I.D.F.) is marked to indicate that it is now attached to a conceptual and therefore must not be used by the system. At the same time the required conceptual line is marked 'attached' in the I.D.F. Pointers in the I.D.F. are set up to connect the actual line with the specified conceptual and the conceptual line number for the line is stored.

ATTACH (AH)

A check is then made as to whether the conceptual has already been **ONLINED** to an object program. If not, the **ATTACH** command successfully terminates and one of the following messages is output to the operator's console:

UNIT *n* LINE *m* ATTACHED TO LINE *p* OF CONCEPTUAL *z* (if the line being attached is a multiplexer line)

UNIT *n* ATTACHED TO LINE *p* OF CONCEPTUAL *z* (if a uniplexer is attached to the conceptual)

If the conceptual is already **ONLINE**, then the newly attached line is introduced to the object program and the same messages as above are output on the operator's console.

Examples

ATTACH U29.L3,CL FRED.L2

Attaches line 3 of unit 29 to line 2 of conceptual FRED.

ATTACH U7,CL FRED.L5(3)

Attaches the uniplexer, unit 7, to the first one of lines 5, 6 or 7 of conceptual FRED that is free.

ATTACH U60.I3,CL BILL.I(2, 5, 7)

Attaches identifier 3 of unit 60 to the first of the identifiers 2, 5 or 7 of conceptual BILL that is free.

ATTACH CL BILL

Attaches the remote operator's console from which the command was issued to an identifier of conceptual BILL.

Error messages

LINE (or IDENTIFIER) *z* IS BUSY

NO LINES (or IDENTIFIERS) AVAILABLE ON THIS CONCEPTUAL

THIS CONCEPTUAL DOES NOT EXIST

THIS LINE (or IDENTIFIER) IS ALREADY ATTACHED

THIS COMMAND IS NOT ALLOWED VIA THIS TYPE OF CONNECTION

THIS UNIT IS ALREADY ATTACHED

TOO MANY PARAMETERS

USER NAME/PASSWORD INVALID

UNIT *z* IS NOT SUITABLE FOR SYSTEM USE

UNIT *z* IS BUSY

UNIT *z* IS NOT A MULTIPLEXER OR UNIPLEXER

z IS NOT A VALID CONCEPTUAL NAME

z IS NOT A VALID LINE NUMBER

CANCELLATION OF ATTACHMENTS

Function

Cancels the attachment of the specified line or identifier of a multiplexer or a uniplexer to a conceptual.

Format

- 1 For character buffering systems:

CANCEL ATTACH *unit number.line number*

The line number is only applicable to multiplexers and should not be used when referring to a uniplexer.

- 2 For message buffering systems:

CANCEL ATTACH *unit number.identifier*

ATTACH (AH)

Execution

The specified line or identifier will be detached from the conceptual named in the ATTACH command. When the CANCEL ATTACH command has been implemented, the line or identifier will return either to MOP or other system use, as appropriate.

Error messages

LINE z IS BUSY	}	For a character buffering multiplexer
LINE z IS NOT ATTACHED		
UNIT z IS BUSY	}	For a uniplexer
UNIT z IS NOT ATTACHED		
IDENTIFIER z IS BUSY	}	For a message buffering multiplexer
IDENTIFIER z IS NOT ATTACHED		

ATTRIBUTE (AU)

Function

Connects a peripheral property with a particular unit on the installation.

Format

ATTRIBUTE *peripheral description*,PR *property name*

Forbidden contexts

NOT OPERATOR

Execution

Checks are made that the property name specified is known to the system and is not the special property CENTRAL; that the unit referred to exists; and that five attributions have not already been made to that unit. Then, if the property specified is not a console property, it will be ATTRIBUTED to the named peripheral.

If, however, the specified property is a console property, the effect of the command depends on whether the named peripheral is local or remote. If it is a local peripheral, a command error is generated. If it is a remote peripheral, a check is made that the device concerned does not already have a console property ATTRIBUTED to it. If it does not, the property is ATTRIBUTED; otherwise a command error is generated.

Examples

ATTRIBUTE 14,PR GASBILLS
AU U14,PROPERTY GASBILLS

AU U29.L2,PR MOP

AU U29.L3.IAA2,PR OXFORD

These have the same effect. Unit 14 must be a basic output peripheral; GASBILLS must have been declared by a PROPERTY command.

This attributes the property MOP to the MOP terminal on line 2.

This attributes OXFORD (possibly a console property) to the terminal peripheral on interface 2 of the 7020 terminal known as AA.

Error messages

PROPERTY NAME PARAMETER MISSING

UNIT *n* MAY NOT BE GIVEN A PROPERTY

UNIT *n* ALREADY HAS 5 ATTRIBUTIONS

PROPERTY NAME *z* UNKNOWN

UNIT GEOGRAPHICAL NUMBER PARAMETER MISSING

UNIT *n* MAY NOT BE GIVEN A CONSOLE PROPERTY

PROPERTY 'CENTRAL' MAY NOT BE ATTRIBUTED

THIS UNIT ALREADY HAS PROPERTY *z* ATTRIBUTED

For attributions to remote peripherals only:

PARAMETER MISSING

PARAMETER FORMAT ERROR: *n*

7020 TERMINAL/INTERFACE NOT DEFINED IN IDF

UNIT *n* ALREADY HAS A CONSOLE PROPERTY ATTRIBUTED

z IS NOT A CORRECTLY FORMED NAME

ATTRIBUTE (AU)

CANCELLATION OF ATTRIBUTIONS

The attribution of a property to a peripheral is removed by the CANCEL ATTRIBUTE command.

Format

CANCEL ATTRIBUTE *peripheral description*,PR*property name*

The specified property must have been ATTRIBUTED to the unit specified. The *property name* may not be the special property CENTRAL.

Forbidden contexts

NOT OPERATOR

Execution

The attribution of the property to the peripheral is cancelled. If the property was EXCLUSIVE and there are no other EXCLUSIVE properties ATTRIBUTED to this unit the peripheral will be available for general use.

Examples

CANCEL ATTRIBUTE 14, PR GASBILLS

CC AU U29.L3.IAA2,PR OXFORD

Error messages

PROPERTY NAME *z* UNKNOWN

PROPERTY *z* WAS NOT ATTRIBUTED TO UNIT *n*

PROPERTY NAME 'CENTRAL' MAY NOT BE ATTRIBUTED

PARAMETER MISSING

PARAMETER FORMAT ERROR: *n*

BROADCAST (BC)

Function

Outputs a given text on all or a selection of the MOP consoles from which user jobs are being run.

Format

- 1 BROADCAST *text*, ALL
- 2 BROADCAST *text*, *job name*, *user name*
- 3 BROADCAST *text*, *job number list*

In format 1 ALL is optional and is assumed if omitted. The text may be up to 50 characters long: excess characters are lost.

Forbidden contexts

NOT OPERATOR, REMOTE

Execution

The text is output on the appropriate MOP console for each MOP job referred to by the parameters. The format of the output is as follows:

time BROADCAST: *text*

A copy of the message is sent to the job's monitoring file (if it exists) in the BROADCAST category.

Note: Jobs run by anticipated command issuers are classed as MOP jobs, and messages may therefore be BROADCAST to them (see Chapter 5 of the manual *Operating Systems GEORGE 3 and 4*).

Error messages

NO PARAMETERS

USER NAME PARAMETER MISSING

NO MOP JOBS IN THE SYSTEM

ERROR *z*, USER NAME IS NOT A CORRECTLY FORMED NAME

JOB *z* IS NOT MOP

NO SUCH JOB NO. AS *z*

JOB NO. *z* IS NOT MOP

NO SUCH JOB AS *z*

If one of the numbers in *job number list* is illegal and an error message is output, the BROADCAST command will continue to analyze the other numbers. Hence several error messages may be output.

CANCEL (CC)

Function

Cancels a previously issued ALLOCATE, ARCHIVE, ASSOCIATE, ATTACH, ATTRIBUTE, CONCEPTUAL, ENTRUST, FINISH, PROPERTY or RETRIEVE command.

Format

CANCEL *command name* *command parameters*

The *command name* can be one of those listed in *Function*, above, or its equivalent two letter code and the *command parameters* are the parameters, if any, associated with the command being cancelled. There must be at least one space between the *command name* and *command parameters*; if there is more than one parameter, the parameters must be separated by commas.

Forbidden contexts

As for the command being cancelled.

Execution

The effect of the command will depend on the command being cancelled. Full details are given under the specification of the appropriate command.

Error messages

The following messages may be given for errors occurring in any of the versions of the CANCEL command; for error messages which occur only with one version see the error message section of the CANCEL section of the appropriate command.

z MAY NOT BE CANCELLED

THIS COMMAND MAY NOT BE CANCELLED IN z CONTEXT

COMMAND PARAMETER MISSING

COMMAND PARAMETER NULL

VERB TOO LONG

CANTDO (CD)

Function

Informs GEORGE that the operator is unable to implement a system request.

Format

CANTDO *system request*

The parameter is enough of an extant system request to be unique. System requests are messages output on the operator's console that may or may not begin with the word PLEASE. For example, if either

PLEASE LOAD *MT 70310

or

LOAD *MT 70310

is a system request, the corresponding CANTDO command for both requests is

CANTDO LOAD *MT 70310

Spaces are not significant in the parameter of the CANTDO command.

Forbidden contexts

NOT OPERATOR

Execution

If the parameter of the command does not identify an extant system request, the error message

z IS NOT AN EXTANT REQUEST

is output on the operator's console.

Otherwise GEORGE will take the alternative action that is standard for the activity that generated the request; for example, it may generate a command error or program failure in the job concerned.

If CANTDO is issued in reply to a system request to engage a basic output device, the device will be WRONGed automatically and the following messages will be output at the operator's console:

UNIT *peripheral description* WRONGED

UNIT *peripheral description* FREE

where *peripheral description* is as described on page 66. The device can be made available for use again by issuing a RIGHT command for it.

Example

CANTDO ENGAGE UNIT 21

Note

The CANTDO command should only be used when inability to implement the request is likely to be permanent. If the operator does not take the action requested and does not issue a CANTDO command, GEORGE will wait until he does, periodically repeating the request on the operator's console.

Error messages

PARAMETER TOO SHORT TO IDENTIFY MESSAGE

z IS NOT AN EXTANT REQUEST

MESSAGE PARAMETER MISSING

CARTRIDGE (CG)

Function

Informs GEORGE that the operator is about to move a cartridge which contains at least one filestore or swap residence from one transport or spindle to another.

Format

CARTRIDGE *Unumber, Unumber*

where both parameters are operator's unit numbers. One parameter must specify the transport or spindle on which is loaded the cartridge that contains a residence.

The other parameter specifies the transport or spindle to which the above cartridge is to be moved. Any cartridge on this second transport or spindle must not contain a residence. The order of the parameters is optional.

Forbidden contexts

NOT OPERATOR

REMOTE

This command is only allowed after the GEORGE READY message has been output.

Execution

After the command has been issued and the transports or spindles specified in the command have been disengaged by GEORGE the operator can unload the cartridge which contains a residence or residences and is specified as one of the parameters and load it on the transport or spindle whose operator's number is specified as the other parameter.

When the transport/spindle which should now contain the system cartridge has been re-engaged GEORGE will read the storage unit's control header and check the serial number. Until the correct cartridge has been loaded and engaged on this transport or spindle GEORGE will not perform any transfer to either device. If the correct cartridge is not loaded on its new transport or spindle GEORGE will disengage the device and output:

UNIT *n* DISENGAGED-PLEASE LOAD CARTRIDGE *z* ON UNIT *n*

If the correct cartridge is loaded GEORGE will carry on reading and writing to the residence on its new transport or spindle.

Note

If a filestore residence is moved and GEORGE is not informed in advance by a CG command, the effect is undefined.

Error messages

DEVICES ARE OF INCORRECT TYPE

COMMAND FORMAT IS INCORRECT

COMMAND PARAMETERS REFER TO THE SAME DEVICE

ONE CARTRIDGE MUST CONTAIN A RESIDENCE

CG COMMAND IN PROGRESS

DEVICES HAVE DIFFERENT CHARACTERISTICS

THE CARTRIDGES MUST NOT CONTAIN THE GEORGE FILE

CHANGELIST (CH)

Function

Changes the properties specified in a LISTFILE entry in :SYSTEM.OUTPUT to the properties specified in the CHANGELIST command. If none are specified in the CHANGELIST command, the property record in the LISTFILE entry is deleted and the output is produced on a central device. It is used mainly to enable a LISTFILE to a device on a given cluster to be changed so that the output appears on a different cluster, but may also be used to change ordinary properties. Note that all the properties required in the LISTFILE entry must be respecified if a CHANGELIST command is used as the properties specified in the original LISTFILE command are deleted.

Format

CHANGELIST *user name, job name, output peripheral type, file description, PROPERTY property names*

The property parameter is optional; if it is included, PROPERTY may be abbreviated to PR.

Forbidden contexts

NOT OPERATOR

Execution

The appropriate entry or entries are found in :SYSTEM.OUTPUT using the first four parameters. If there is no property parameter, the properties specified in the :SYSTEM OUTPUT entry are deleted and the LISTFILE is output on a central device. If there is a property parameter, the properties specified replace those in the :SYSTEM.OUTPUT entry.

Notes

If the command is issued after GEORGE has found a suitable peripheral on which to output the listing, the message

REQUIRED FILE BEING IMPLEMENTED ON UNIT *n*

is output and the CHANGELIST command is not implemented.

If the command is issued while GEORGE is searching for a peripheral on which to output the listing, the message

REQUIRED FILE BEING IMPLEMENTED

is output on the operator's console and the CHANGELIST command is not implemented. In this case, if it is still desired to CHANGELIST, then first ensure that GEORGE does not find a suitable peripheral, by replying CANTDO to a PLEASE ENGAGE message or by WRONGING the peripheral; then issue the CHANGELIST command again.

Error messages

PARAMETER MISSING

EXPRESSION INVALID

THERE IS NO SUCH ENTRY AS *z*

PERIPHERAL TYPE *z* NOT AVAILABLE

NOT AVAILABLE DUE TO EXCLUSIVE PROPERTIES

PERIPHERAL PROPERTIES *z* NOT AVAILABLE

MORE THAN ONE CONSOLE PROPERTY GIVEN

MORE PROPERTIES SPECIFIED THAN ALLOWED

PROPERTIES NAME *z* UNKNOWN

CONCEPTUAL (CL)

Function

Introduces a conceptual name to the system and gives that conceptual a maximum number of slots and a lowest slot number. The conceptual character or message buffering multiplexer will be built up using the ATTACH command. If the lowest slot number is greater than 1, a conceptual message buffering multiplexer will be defined automatically by the system and only an identifier of a message buffering multiplexer can be attached to it. Otherwise the named conceptual is not previously defined for use by a specific system, but when any attachment of one kind has been made, subsequent attachments must be of the same kind.

In the case of a message buffering conceptual, slots of the conceptual correspond to identifiers. In the case of a character buffering conceptual, slots of the conceptual correspond to lines.

Format

CONCEPTUAL *conceptual name, numerical expression, username*

The *numerical expression* can take any of the following three forms:

- 1 *n* is the maximum number of slots in the conceptual. The lowest conceptual slot number will be taken as 1.
- 2 *m(n)* where *m* is the lowest conceptual slot number and *n* is the maximum number of slots in the conceptual.
- 3 If omitted, both the lowest conceptual slot number and the maximum number of slots will be taken as 1.

This is needed to determine how many slots the conceptual should be defined to have and how these slots will be numbered.

Note that the number of lines in a character buffering conceptual must be in the range 1 to 63 and the highest permissible identifier number in a message buffering conceptual is 4095.

The *conceptual name* consists of up to 12 alphanumeric characters, spaces or hyphens, with the first character alphabetic.

The *username* parameter is optional. Its use is described below.

Forbidden contexts

NOT OPERATOR, REMOTE

Execution

The *conceptual name* and *numerical expression* are checked for legality and a check is made to see if the conceptual name is already defined in the Installation Description File. If it does not exist, a new conceptual with a defined number of slots and a lowest slot number is created. If a user name is present, it is checked for legality and an entrant with the same name as the conceptual is set up under the directory of the user specified. The user may then set traps on the entrant for other users.

If he wishes to allow another user to ONLINE his conceptual, he must set up a WRITE trap for that user, and if he wishes to allow another user to ATTACH a MOP terminal to his conceptual, he must set a READ trap for the user concerned.

If the conceptual does already exist in the I.D.F. and is not ONLINEd, several checks are made and an action, based on their results, is taken:

- 1 If the new lowest slot number is different from the old one, a command error is reported.
- 2 If the new number of slots is different from the old one, the size of the conceptual is adjusted according to the new value.
- 3 If the conceptual is already owned by a user and a new user name is specified, a command error is reported.
- 4 If the conceptual exists without an owner, the ownership is established as described above.

If the conceptual already exists and is ONLINE to a program, a command error is reported.

Examples

CONCEPTUAL DICK,5

This will cause a conceptual named DICK to be set up with five slots numbered from 1 to 5

CONCEPTUAL FRED,4(10),:MANAGER

This will cause a conceptual named FRED to be set up, with 10 slots which will be numbered from 4 to 13. The ownership by :MANAGER will be established.

Error messages

PARAMETER MISSING

EXPRESSION INVALID

number IS OUT OF RANGE

conceptual name IS ONLINED

user name IS NOT A CORRECTLY FORMED NAME

conceptual name HAS ALREADY BEEN USED FOR ANOTHER FILE (IN USER'S DIRECTORY)

CANCELLATION OF CONCEPTUALS

The *conceptual name* is deleted from the system by the CANCEL CONCEPTUAL command.

Format

CANCEL CONCEPTUAL *conceptual name*

Forbidden contexts

NOT OPERATOR, REMOTE

Execution

The conceptual name is deleted from the system and all attached lines are implicitly detached.

Example

CANCEL CONCEPTUAL FRED

Error messages

conceptual name IS ONLINED

EXPRESSION INVALID

conceptual name IS NOT A CONCEPTUAL NAME

DEAD (DD)

Function

Withdraws magnetic tapes from the system.

Format

DEAD *tape serial number*₁, *tape serial number*₂, . . . , *tape serial number*_n

The tape serial numbers must be qualified by *MT.

Execution

If a specified tape is a pool tape or work tape, then its record is deleted from the index file :SYSTEM.SERIAL, provided that it is not in query status. All information concerning the tape is thus removed from the system.

All the tapes described by the parameters will be dealt with except those for which error messages are output. If the tape is in use or being waited for, it does not become insecure until the users have completed their access to it.

Example

DEAD (360245) (*MT), (360247) (*MT)

Error messages

DEVICE TYPE QUALIFIER IMPERMISSIBLE

DEVICE TYPE QUALIFIER MISSING

MAGNETIC TAPE DESCRIPTION FOR THIS COMMAND MUST CONSIST OF A SERIAL NUMBER ONLY

MT *z* IS IN QUERY STATUS

SERIAL NUMBER ALREADY GIVEN IN THIS COMMAND

SERIAL NUMBER MISSING

z IS AN OWNED TAPE

z IS A XENOTAPE AND MAY NOT BE REFERRED TO BY THIS COMMAND

z IS NOT A CORRECT ENTRANT DESCRIPTION FORMAT

(This message may be preceded by one of a number of supplementary messages, which are self-explanatory.)

z IS NOT KNOWN TO THE LIBRARIAN

DISENGAGE (DG)

Function

Terminates the reading of commands from a paper tape or card reader.

Format

DISENGAGE

Forbidden context

NOT READER

Although this command may not be issued from the operator's console, but only from a paper tape or card reader, it is included in this chapter because an operator may need to know about it, for example to correct an error in the card containing the command.

Execution

The reader from which the command is issued is disengaged. It remains allocated to GEORGE and can subsequently be re-engaged in the normal way. If the reader is on a 7020 terminal, the device is WRONGed and will need to be RIGHTed by the operator before more data can be read.

Error messages

None.

DOCUMENT (DOCU or DM)

Function

Introduces an external document on a basic peripheral.

Format

DOCUMENT *document name*

The *document name* may be given in either the full or the short form, but if the command which subsequently calls for the document uses the full form, the *document name* must be given in full.

A *short document name* consists of up to twelve letters, digits, spaces or hyphens starting with a letter.

A *full document name* is a username followed by a short document name.

Forbidden context

NOT READER, REMOTE

Although this command may not be issued from the operator's console, but only from a paper tape or card reader, it is included in this chapter because an operator may need to know about it, for example to correct an error in the card containing the command.

Execution

The document heading is stored in a file. When a command is given to read the document, GEORGE finds the heading and allocates the peripheral on which the document has been loaded. If the document has not been loaded the following message is output on the operator's console:

PLEASE LOAD DOCUMENT *document name*

This message is output at regular intervals until the document is loaded.

Examples

DOCUMENT INPUTDATA

INPUTDATA is a *short document name*

DOCUMENT :FRED.CARDDATE

FRED.CARDDATE is a *full document name*

Error messages

DOCUMENT NAME FORMAT ERROR

DOCUMENT NAME PARAMETER MISSING

DUMP (DU)

Function

- 1 Starts the incremental dumper.
- 2 Starts the dump processor.
- 3 Releases obsolete dump tapes for re-use by the dumper.
- 4 Returns obsolete dump tapes to the pool.

Format

- 1 To start the incremental dumper:

DUMP

or

DUMP TOTAL

- 2 To start the dump processor:

DUMP PROCESS

- 3 To release obsolete tapes for re-use by the dumper:

(a) DUMP REUSE,ALL

(b) DUMP REUSE,*number*

(c) DUMP REUSE,*tsn*₁(*MT),*tsn*₂(*MT),...*tsn*_{*n*}(*MT)

where *number* is the number of tapes to be released for re-use, and *tsn* is the tape serial number of a tape to be released for re-use.

- 4 To return obsolete dump tapes to the pool:

(a) DUMP RETURN,ALL

(b) DUMP RETURN,*number*

(c) DUMP RETURN,*tsn*₁(*MT),*tsn*₂(*MT),...*tsn*_{*n*}(*MT)

where *number* is the number of tapes to be returned, and *tsn* is the tape serial number of a tape to be returned.

Forbidden contexts

NOT OPERATOR

Execution

- 1 The first form of the command can be used to override the installation parameter DUMPTIME which determines when the incremental dumper is to be started. The DUMP command causes the dumper to start within one minute. If the parameter TOTAL is present, the entire filestore will be dumped, except for files that are open for writing or not on-line. A warning about the use of this parameter is given in Chapter 4 of *GEORGE 3 and 4 Operation Management*.
- 2 The second form of the command should be used to start the dump tape processor, which can only be started by this command.
- 3 If the third form of the command is used the specified dump tapes (Format 3(c)), or the specified number of dump tapes (Formats 3(a) and 3(b)), provided that they are obsolete, are released for re-use by the dumper. The effect is that when the dumper requires a new tape it will use the tapes released by this command before requesting a pool tape.

If Format 3(a) is used, all tapes that are obsolete are released for re-use by DUMPER. In Format 3(b), if fewer tapes are obsolete than are specified in the parameter, only those that are obsolete are released.

DUMP (DU)

The following message is output at the operator's console:

MTS *list of tape serial numbers* TO-BE-REUSED – INCREMENTS OBSOLETE

- 4 If the fourth form of the command is used the specified obsolete dump tapes (Format 4(c)), or the specified number of obsolete dump tapes (Formats 4(a) and 4(b)), provided that they are obsolete, are returned to the pool. If Format 4(a) is used, all tapes that are obsolete are returned. In Format 4(b), if fewer tapes are obsolete than are specified in the parameter, only those that are obsolete are returned.

The following message is output at the operator's console:

MTS *list of tape serial numbers* RETURNED INCREMENT(S) OBSOLETE

One or more messages as follows (saying which increments are obsolete on which tapes) may be output:

INCREMENT(S) *list of increments* OBSOLETE ON MT(S) *list of tape serial numbers*

Also, if all the tapes for an increment have been released, that increment must have been obsolete and so its record in INCINDEX is deleted. If any increment is deleted, the following message is output:

RECORD(S) DELETED FOR INCREMENT(S) *list of increments*

Examples

DUMP

DUMP RETURN,20

DUMP REUSE,(60533)(*MT)

A normal incremental dump will be initiated

20 obsolete tapes (or all obsolete tapes if there are fewer than 20) will be returned to the pool.

Tape 60533 will be marked 'for reuse by dumper', provided it is obsolete.

Error messages

z NOT RECOGNISED

EXPRESS (EP)

Function

- 1 Informs the high level scheduler subject program that a particular job is of great urgency and should not be made to wait before being made fully started.
- 2 Communicates a message to the high level scheduler subject program, which will cause it to perform one of a number of tasks.

Format

- 1
 - (a) EXPRESS *job number*
 - (b) EXPRESS *jobname,username*
- 2
 - (a) EXPRESS *job number,text*
 - (b) EXPRESS *jobname,username,text*

In Formats 1(a) and 2(a) *job number* is the number that the system uses to identify a particular job. The operator can determine this number by means of a WHATJOBS command.

In Formats 1(b) and 2(b) the *jobname* and *username* parameters are interchangeable.

In Formats 2, the *text* must be one of a set of character strings known to the high level scheduler, or the command will be ignored. The *text* may not be longer than 24 characters; excess characters are truncated.

In Format 2(a), the first parameter may be null. Thus in cases where the *text* is not applicable to a particular job, it is not necessary (although it is still possible) to refer to a job number.

Forbidden contexts

NOT OPERATOR

Execution

FORMAT 1

The command checks whether the specified job exists and reports an error if it does not. The EXPRESS command has no effect for a system issued job; if it is issued for such a job the following message is output to the operator's console:

EXPRESS COMMAND IS NOT MEANINGFUL FOR A SYSTEM ISSUED JOB

If the specified job exists and is not system issued, the high level scheduler subject program is informed that an EXPRESS command has been issued for this job and one of the following messages is output to the operator's console, depending on which format of the command was used.

If Format (a) was used:

EXPRESS COMMAND FOR JOBNUMBER *x* ACKNOWLEDGED

If Format (b) was used:

EXPRESS COMMAND FOR *jobname, username* ACKNOWLEDGED

The job is given a very high CPI and other non-expressed jobs are given a zero CPI. The job is given UR A when it becomes fully started. No budget checking is done. If the job is already fully started, then the urgency cannot be reset unless the job becomes tentatively started and then fully started again.

If the EXPRESS command is cancelled, then the job will have its urgency allocated in the usual way.

For further details of how the high level scheduler makes jobs fully started see Chapter 8 of the manual *GEORGE 3 and 4 Operation Management*.

FORMAT 2

If the *text* is one of a set of character strings known to the high level scheduler, then appropriate action is taken; otherwise the command is ignored. The character strings for the standard ICL high level scheduler are defined in Chapter 8 of the manual *GEORGE 3 and 4 Operation Management* and they instruct the HLS to perform certain tasks. These tasks include BREAK, SWITCH, CANCEL EXPRESS, allow/inhibit job saving, allow/inhibit budget checking.

EXPRESS (EP)

If a job is specified in the command it will *not* be expressed unless the *text* indicates that this is to be done. With the standard ICL high level scheduler a text of EP means that the job is to be expressed.

The standard ACKNOWLEDGED message is output for both formats 2(a) and 2(b) except that if the first parameter is null in format 2(a) the message is:

EXPRESS COMMAND ACKNOWLEDGED

Examples

Command

EXPRESS 3

EP FRED,:JONES

EXPRESS :JONES,FRED

Resulting message

EXPRESS COMMAND FOR JOBNUMBER 3
ACKNOWLEDGED

EXPRESS COMMAND FOR FRED, :JONES
ACKNOWLEDGED

EXPRESS COMMAND FOR FRED, :JONES
ACKNOWLEDGED

Error messages

z DOES NOT EXIST

NO SUCH JOBNUMBER AS z

z IS NOT A CORRECTLY FORMED NAME

z IS NOT A VALID NUMBER FORMAT

JOBNAME PARAMETER MISSING

USERNAME PARAMETER MISSING

PARAMETER MISSING

Notes

- 1 The EXPRESS command should not be used with system issued jobs since these are made fully started automatically.
- 2 The command makes no checks on the state of the job. Thus, if the specified job is already fully started, this will not be detected and the standard ACKNOWLEDGED message will be output.

FINISH (FN)

Function

Prevents the system from accepting or starting any more jobs. If a parameter is given, existing jobs will be finished immediately or within the time specified.

Format

1 FINISH NOW

2 FINISH *length of time*

where *length of time* is *number*MINS, *number*SECS, or *number*HRS, and must not exceed 3 days (that is, maximum values are 72HRS, 4320MINS or 259200SECS). If no units are given, MINS is assumed.

3 FINISH

Forbidden contexts

NOT OPERATOR, REMOTE

Execution

No more jobs will be accepted by the system and a user who tries to introduce a new job will receive the message:

NO MORE JOBS ARE BEING ACCEPTED: SYSTEM CLOSING DOWN

{	NOW (Format 1)
	IN <i>time</i> (Format 2)
	SOON (Format 3)

Further action taken depends on the format, as follows:

FORMAT 1

The High Level Scheduler program will be informed so that it may take appropriate action and no more jobs will be made fully started. Any jobs already in the system will be abandoned. The message:

ABANDONED : SYSTEM CLOSING DOWN

is output to the MOP terminal and/or the monitoring file. All LISTFILES and INPUTs in progress will be abandoned. When all jobs (including those issued by the system) have finished or been abandoned, the message

ALL USER JOBS NOW FINISHED

is output on the operator's console.

FORMATS 2 AND 3

All MOP jobs in progress will be informed with the message

SYSTEM CLOSING DOWN

{	IN <i>time</i> (Format 2)
	SOON (Format 3)

The High Level Scheduler program will be informed and no more jobs will be made fully started. Tentatively started jobs will be run until they have finished or require to be made fully started. In this case background jobs will be abandoned with the message

ABANDONED: SYSTEM CLOSING DOWN

and MOP jobs will receive the message:

YOU MAY NOT BE MADE FULLY STARTED

Fully started background jobs will be allowed to continue until either they have finished or (in format 2) the time given as a parameter has elapsed, when FINISH NOW action will be taken for them. Background jobs and MOP jobs in limbo may be connected until the *time* specified expires. LISTFILES waiting to be output can be initiated provided (in format 2) that this takes place before the time given has passed. At the end of this time any LISTFILES and INPUTs in the above categories will be allowed to finish. LISTFILES to the monitoring file and INPUTs in USER context will be abandoned when the job issuing them is abandoned.

FINISH (FN)

When all background jobs have finished, the message

ALL BACKGROUND JOBS FINISHED: *n* MOP JOBS LEFT

is output on the operator's console (provided it is before *time* in format 2). When all jobs, LISTFILES and INPUTs have finished, the message

ALL USER JOBS NOW FINISHED

is output.

In all three formats, when the FINISH command has completed all it has to do (that is, everything has been terminated), the message

EVERYTHING HAS FINISHED

will be output and the system can be deleted.

Notes

- 1 A user's job may take its own FINISH action. If the job has previously issued a WHENEVER FINISH command then the command given in the second half of the WHENEVER command will be obeyed. Such jobs can also test whether a FINISH command has been issued by means of the IF FINISH command. Note that this condition is unset by the CANCEL FINISH command (see below).
- 2 System issued jobs must take their own FINISH action by means of the WHENEVER FINISH and IF FINISH commands.
- 3 The length of time may be extended or shortened by issuing another FINISH command provided that this occurs before FINISH NOW action has been taken.
- 4 The effect of a FINISH command may be cancelled by use of the CANCEL command (see below).
- 5 Any jobs which have been saved by the high level scheduler will remain saved over the system close down if there is no general restore at E.M.S.

Error messages

PARAMETER FORMAT ERROR

FINISH NOW ACTION ALREADY STARTED

CANCELLATION OF FINISH

Function

Cancels the effect of a FINISH command.

Format

CANCEL FINISH

Forbidden contexts

NOT OPERATOR

Execution

Provided that FINISH NOW action has not already been initiated, the ban on accepting jobs will be lifted and they may be introduced subject to the usual JOBLIMIT restrictions. Any MOP jobs still running will be informed of the CANCEL with the message:

SYSTEM CLOSE DOWN HAS BEEN CANCELLED

Any jobs already abandoned must be restarted by the operators. Any LISTFILE or INPUT commands not permitted after FINISH must be reissued.

FINISH (FN)

Error messages

ONLY ALLOWED AFTER FINISH

FINISH NOW ACTION ALREADY STARTED

z IS TOO LONG A TIME

HARDWARE (HW)

Function

Introduces to the system details of the devices attached to lines of character buffering uniplexers or multiplexers.

Format

HARDWARE *Unumber.Ln(m).description*

where *number* is the peripheral unit number

Ln(m) specifies a group of *m* lines starting at *n*; if (*m*) is omitted, then *Ln* specifies a group which consists of a single line, indicated by *n*

description is a description of the terminal attached to the line being described. This parameter can be one of:

MOP For standard MOP devices

TER (*des*) For 7020 terminals and 1901A remote job entry terminals. For a 7020 terminal the format of (*des*) is:

(identifier,Tnumber,D(description))

where *identifier* is a two character alphanumeric code that the user defines for each of his terminals and GEORGE uses in communications about the terminal. The first character must be an alphabetic character. The manager must ensure that each *identifier* is unique.

number is the number given to the terminal by the engineers and is in the range 1 to 12 inclusive.

description is a string of six parameters that describe the peripheral on each interface. The peripherals are described by TW (typewriter), TR, TP, CR and LP. The first parameter specified is implicitly on interface 1, the second on interface 2, and so on. It is therefore very important to define the interfaces in the correct order, otherwise messages could be directed to the wrong peripherals.

For 1901A remote job entry terminals the format of (*des*) is the same as for 7020 terminals except that 16 is added to the engineers' terminal number in the *number* parameter, giving a number in the range 17 to 28.

SEN For lines that are connected to sentinels

NUL For lines that are not to be used by GEORGE

Forbidden contexts

NOT OPERATOR, REMOTE

Execution

If a new terminal description is being introduced to a line, and the line is not NUL, but is TER (that is, the line currently has a 7020 on it), and the new terminal is a 7020, then the new 7020 will be added so long as no terminal with an identical engineers' terminal number has already been defined to exist on the line. Thus 1901A RJE terminal number 20 cannot co-exist with 7020 terminal number 4 on the same line.

The peripheral descriptions are checked to ensure that positions 1, 3 and 5 in the parameter describe input devices and positions 2, 4 and 6 describe output devices. Each peripheral description must include one typewriter input and one typewriter output.

The NUL description will only be allowed for lines for which no terminal or terminal peripheral is a cluster member or a cluster console. Thus cluster members and cluster consoles should be cancelled by use of the appropriate forms of the ATTRIBUTE and PROPERTY commands before a HARDWARE command with NUL description can be given for such lines. The NUL description will cause all other attributions and all associations for the line to be cancelled.

It is not possible to use the HARDWARE command to redefine a line connected to a unit that is MOPped ON unless either the line was previously defined as NUL or the line is currently ATTACHED to a conceptual multiplexer. When the HARDWARE command is used to redefine an unattached NUL line on a unit that is MOPped ON the new device will be started up immediately.

If this command is used to re-define lines of a multiplexer or the line of a uniplexer the new definition will not be noticed until GEORGE needs to inspect the IDF for that line. This inspection will occur for one of three reasons:

- 1 A MOP ON command has been issued for the unit
- 2 A line which was previously attached to a conceptual multiplexer/uniplexer has been detached, or the conceptual has been released
- 3 A break-in signal has been noticed on an idle line (MOP only).

Examples

```
HW U30.L1,MOP
HW U30.L2,SEN
HW U30.L3,TER(IAA,T1,D(TW,TW,CR,LP,TR,TP))
HW U30.L3,TER(IAB,T2,D(TW,TW,,LP))
HW U30.L4(9),MOP
HW U30.L3,TER(IAC,T19,D(CR,LP,TW,TW))
```

Error messages

PARAMETER MISSING

PARAMETER FORMAT ERROR: *x*

ILLEGAL DEVICE ON INTERFACE *x*

REMOTE CONSOLE MUST BE DEFINED FOR BOTH INPUT AND OUTPUT

INVALID TERMINAL NUMBER *n*

x IS NOT A VALID NUMBER FORMAT

CLUSTER CONSOLES(S) DEFINED ON LINE *x*

DEVICE(S) WITH CONSOLE PROPERTIES ATTRIBUTED EXIST ON LINE *x*

NO UNITS IN INSTALLATION DESCRIPTION FILE

7020 TERMINAL NUMBER *n* ALREADY DEFINED

DEVICE ALREADY DEFINED ON LINE *n*

IDFLIST

Function

IDFLIST is a system macro which lists the contents of the Installation Description File (IDF) on a line printer.

Format

IDFLIST

Execution

The macro creates a copy of the IDF in printable form and then issues a LISTFILE to produce a listing of this file. When the listing is completed the file is erased.

Note that it is only possible to issue one IDFLIST at a time. If this rule is not observed, the error message below is output.

Error messages

| JOBNAME IDFLIST NOT UNIQUE

INPUT (IN)

Function

Reads the lines immediately following into a basic file.

Format

INPUT *username, file description, TABS(P₁,P₂,...P_n) terminator*

The TABS(P₁,P₂,...P_n) and *terminator* parameters are optional and may be submitted in either order.

The TABS(P₁,P₂,...P_n) parameter may be specified with the P₁,P₂,...P_n indicating the character positions of the tabulations used in a fixed format language.

If the INPUT command is on paper tape, the *file description* may include a type qualifier (GRAPHIC, NORMAL or ALLCHAR) to indicate the type of file to be created.

See Chapter 10 of *Operating Systems GEORGE 3 and 4* for definitions of *file description* and *terminator*.

For example for a PLAN source program the TABS parameter would be

TABS(7,13,16,36,73,76)

If the local filename is to be issued as a macro (by RUNJOB), it must not contain spaces.

Forbidden contexts

OPERATOR,USER

Although this command may not be issued from the operator's console, it is included in this chapter because an operator may on occasion need to know about it, for example to correct an error in the card containing the command.

Execution

The records following the INPUT command are written to the named file (if it exists, otherwise one is created) until a record starting with the four specified terminating characters is encountered.

If the command is issued from a card or paper tape reader and the correct terminator is not found, GEORGE will read to the end of the cards or paper tape and output the message

PLEASE ENGAGE UNIT *number*

on the operator's console. The operator has three alternatives:

- 1 He can reload and engage the peripheral.
- 2 He can issue a CANTDO command, which will erase the file.
- 3 He can issue a TERMINATE command, which will terminate the input and close the file as if a terminator had been read.

These three options are also available to the operator in the event of a peripheral failure. If the failure is caused by a transfer error the message

PLEASE ENGAGE UNIT *peripheral description* FOR ERROR RESTART

will be output.

If INPUT detects that the input file has been filled, or, in the multifile case, the maximum number of elements has been reached, the peripheral will be disengaged and the message:

FILE *filename* OVERFLOWED DURING INPUT/JOB ON UNIT *n*

is output to the operator's console. The operator then has three courses of action:

- 1 He can engage the peripheral in which case INPUT will search for the terminator ignoring records in between.
- 2 He can TERMINATE which will result in the input being terminated normally.

INPUT (IN)

- 3 He can CANTDO which will terminate the input and erase the file or, if the file was being appended to, leave the file as it was before the INPUT command.

Examples

INPUT :PAR,JOHN

The data that follows the command will be written to a graphic file called JOHN belonging to the user :PAR. The lines of data are terminated by the standard terminator, a line beginning ****, which is stored, followed by a blank record.

INPUT :FRED,DATAFILE(GRAPHIC),TABS
(10,20,50),T####

The data will be written to a GRAPHIC file called DATAFILE, belonging to :FRED. The lines of data are terminated by a line beginning ####, which is not to be stored. Any tabulate characters in the data will be converted into the appropriate number of spaces to put the next character into the 10th, 20th, or 50th position—whichever is next.

INPUT :FRED,;Z.CARDFILE

The data will be stored in a file called CARDFILE under :Z's directory where :Z is a pseudo user whose directory belongs to :FRED. The type of the file created depends on the source of the command. The lines of data are terminated by the standard terminator ****, which is stored, followed by a blank record.

Error messages

FILE NAME PARAMETER MISSING

USER NAME PARAMETER MISSING

z IS NOT A VALID TERMINATOR

AN ILLEGAL QUALIFIER HAS BEEN GIVEN

INSTPARA (IP)

Function

The command may have one of three formats listed under *Format*, below. The functions of the command in each case are as follows:

- 1 Lists the maximum, minimum and current value of each Installation Parameter on the operator's console.
- 2 Reports the current value of one of GEORGE's dynamically variable Installation Parameters.
- 3 Sets the value of one of GEORGE's Installation Parameters.

Format

- 1 INSTPARA ALL
- 2 INSTPARA *parameter name*
- 3 INSTPARA *parameter name, new value*

The *parameter name*, which has the format of a local file name, must be one of the names listed in the table below or a name previously specified in an IPEDIT command.

The *new value* must be in standard number format, unless *parameter name* is one of the following:

For the installation parameters JOBTRACE, MOPTRACE, OPTRACE and MINTRACE the *new value* parameter is in the standard form for the *action on monitoring* file parameter list.

For the installation parameter CONTEXT, the *new value* parameter is A, B or C, or any combination thereof.

INSTALLATION PARAMETERS

The following table lists the installation parameters in alphabetical order, giving the initial setting of each parameter with its maximum and minimum values and the units used. The function of each parameter is described briefly; for a fuller description of the meaning and uses of these parameters, see the manual *GEORGE 3 and 4 Operation Management*.

EMS means that the value is calculated at Early Morning Start (this means when GEORGE is loaded): for details see Chapter 4 of the manual *GEORGE 3 and 4 Operation Management*. Other values given are those determined by the issued initial filestore, or, where an installation parameter is created for a new Mark, those set up when the new Mark is first used. The *Initial value* is changed by the INSTPARA command, and the *Max*, *Min*, and *Parameter name* may be changed by the IPEDIT command.

The set of monitoring file categories known as ALL in Mark 6 is now called FULL. The new ALL includes four new categories: CENTRAL, CLUSTER, ENGINEER and JOURNAL.

<i>Parameter name</i>	<i>Initial value</i>	<i>Max.</i>	<i>Min.</i>	<i>Units</i>	<i>Meaning</i>
BACKGROUND	EMS	200	0	Jobs	Maximum number of background jobs which may be fully started.
BACKJAM	80	99	50	%	The proportion of a backing store residence that must be occupied before the backing store unjammer takes action on the residence.
BACKTHRESH	10	50	0	%	The proportion below the value of BACKJAM at which the threshold of all non-special residences is set.
BSINTERVAL	2048	2 ²³ -1	0	Mins.	The target interval between entries to the backing store unjammer.
CHAPTERQUOTA	EMS	EMS	EMS	Words	The chapter quota
CONTEXT	0	A,B,C	0	Context bits	The context privileges that are currently in force.

INSTPARA (IP)

<i>Parameter name</i>	<i>Initial value</i>	<i>Max.</i>	<i>Min.</i>	<i>Units</i>	<i>Meaning</i>
COREOBJECT	EMS	EMS	EMS	Words	The maximum core that can be given to an object program by CORE, SIZE, LOAD, GIVE/4 etc. In GEORGE 3 the value of this parameter is limited by the core size of the machine. In GEORGE 4 it is limited by the amount of backing store reserved to hold object programs.
CPIRATIO	120	500	100	%	Weighting factor used by HLS to calculate wait times for background jobs of various urgencies waiting to be made fully started.
DATE	10	1440	1	Mins.	The interval between date printouts on the operator's console.
DATINTERVAL	5	8388607	1	Mins.	The intervals at which data is collected by the system performance package (see the manual <i>Operation Management</i>) and stored in the file :SYSTEM.PERFORMANCE.
DEFAULTBS	500	10000	10	K words	The maximum on-line backing store allowed to a job at any one time.
DUMPTAPES	1	3	1	Tapes	The number of copies of filestore dumps taken and maintained.
DUMPTIME	280	2000	0	Mins.	The normal time interval between successive incremental dumps of the filestore.
FORMULA	5000	10 ⁶	0		Estimate, periodically reset by GEORGE, of the severity of backing store unjamming necessary to achieve the BSINTERVAL target set by INSTPARA.
HLSREVIEW	1	20	1	Mins.	Maximum time interval between entries to the High Level Scheduler to review job progress.
HOLDUP	300	10000	1	Secs.	The number of seconds to be added to the fair waiting time for a program which cannot be swapped in and is larger than the program quota.
IDIV	2	100	1		Dividing factor in algorithm used to calculate cost of mill time used. Only relevant if BILA (Built-in Log Analysis) is used.
IMULT	1	100	0		Multiplying factor in algorithm used to calculate cost of mill time used. Only relevant if BILA (Built-in Log Analysis) is used.

<i>Parameter name</i>	<i>Initial value</i>	<i>Max.</i>	<i>Min.</i>	<i>Units</i>	<i>Meaning</i>
IPAGE	1	50	1	Pages	The maximum number of pages that can be repeated if an error restart occurs on a line printer while listing a file.
JOBLIMIT	EMS	EMS	2	Jobs	The maximum number of jobs that may be introduced to the system.
JOBTIME	300	3600	50	Secs.	The default value of the allowed mill time for any job.
JOBTRACE	FULL	ALL	NONE	Monitoring file categories	The initial setting for a background job's TRACEing to the monitoring file.
L	1	$2^{23} - 1$	0	Money units	Minimum charge used in the algorithm that calculates the cost of mill time used. Only relevant if BILA is used.
MAXQUOTA	EMS	EMS	1024	Words	This parameter only exists in GEORGE 4. It is the maximum quota that will be given to an object program.
MESSAGETIME	60	400	1	Secs.	The factor used to regulate the time interval between printouts of system messages on the operator's console.
MINTRACE	LOGGING and BROADCAST	ALL	NONE	Monitoring file categories	The minimum set of categories of monitoring file output which may be requested by the TRACE command.
MOPCPI	50	100	0	%	Target percentage of mill time which should be devoted to MOP job.
MOPLIMIT	JOBLIMIT	200	0	JOBS	Number of MOP jobs which may be made fully started.
MOPTRACE	LOGGING and BROADCAST	ALL	NONE	Monitoring file categories	The initial setting for a MOP job's TRACEing to the monitoring file.
OBJECTQUOTA	EMS	EMS	1	Words	The object program quota (reset by EMS). In GEORGE 3 this parameter controls the amount of core GEORGE uses to hold object programs. In GEORGE 4 the parameter is fixed at GEORGE restore time; its value gives the amount of core dedicated to holding object programs, less an allowance for page tables and page turning.

INSTPARA (IP)

<i>Parameter name</i>	<i>Initial value</i>	<i>Max.</i>	<i>Min.</i>	<i>Units</i>	<i>Meaning</i>
OPTRACE	ENGINEER	ALL	NONE	Monitoring file categories	Categories traced to central operator's console, in addition to messages that are sent in any case.
PERFWAIT	720	8388607	10	Mins.	The interval between each initiation of the performance listing job.
PROGTIME	10	600	1	Secs.	Default value of TIME for object programs (overwritten by any TIME commands in the job).
RELOAD	1	1	0		Controls the action of GEORGE at the end of a GEORGE postmortem.
REPORTTIME	30	8388607	1	Mins.	The interval between each output of the job progress report messages to the System Journal.
SIZEDEFAULT	EMS	EMS	EMS	Words	The maximum size which can be given to a core image by GIVE/4 if no MAXSIZE command has been issued. If the High Level Scheduler is being used, the size for the CORE, SIZE, LOAD, RESUME commands etc. is also limited.
SLOTTIME	250	32K-1	10	Milli secs. per K	Controls relationship between the size of an object program and the length of its time slot.
TIME	1	1440	1	Mins.	The interval between time printouts on the operator's console.
VOLUME	8192	2 ²³ -1	1	PERIs	Default setting of the LIMIT qualifier (the number of PERIs that a program may issue to an ASSIGNED file).
WAITTIME	30	90	0	Mins.	The time for which an unterminated MOP or background job remains in the waiting state before it is terminated by an internally issued ENDJOB.
WAITWEIGHT	120	500	100	%	Factor used by HLS to calculate CPIs of jobs of various urgencies.

Table 11: Installation Parameters

Forbidden contexts

USER, MOP, REMOTE

Execution

A given installation parameter may have different values when running under GEORGE 3 and GEORGE 4 and these values are held in separate files. An INSTPARA command is only effective on the value of the parameter associated with the current system.

If a parameter name is specified (format 2 or 3), GEORGE finds the named parameter in the system file containing the installation parameters and their values (:SYSTEM.PARAMINDEX or :SYSTEM.PARAMINDEXG4).

If a new value parameter is included in the INSTPARA command, the named parameter is given this new value provided that the value is within the range specified in the table above. The new value is output on the operator's console in the form

INSTALLATION PARAMETER '*parameter name*' = *new value*

If no new value parameter is given in the command, the current value of the named parameter is output in the form shown above.

If the ALL parameter is specified (format 1), the required information is output under the headings:

NAME	MAX	MIN	VALUE

on the operator's console.

Examples

INSTPARA COREOBJECT, #40000

Reply (to operator's console)

INSTALLATION PARAMETER

'COREOBJECT' = 16384

INSTPARA JOBLIMIT

Reply (to operator's console)

INSTALLATION PARAMETER

'JOBLIMIT' = 50

The maximum amount of core that can be given to an object program will be set to 16K words.

There is no new value parameter so the current maximum number of jobs that may be introduced to the system will be output on the operator's console.

Notes

- At the time of reloading after a GEORGE break, the value of each installation parameter will be the same as it was before the break, provided that:
 - The value is a valid value
 - If a general restore is done at reloading time, any change made by the user of an INSTPARA command occurred before the last dump of the filestore.
 Otherwise, the installation parameters are set to their most recent valid value.
- Further descriptions of the meanings and use of these parameters are given in the manual *GEORGE 3 and 4 Operation Management*.
- No installation parameter can be given the name ALL.
- Whenever the maximum, minimum or current value of the Installation Parameters JOBTRACE, MINTRACE or MOPTRACE is output, it will be in the form:
action on Monitoring File
- Whenever the maximum, minimum or current values of the installation parameter CONTEXT are output, they will either specify one or more of A, B or C or be zero.

Error messages

INSTALLATION PARAMETER *z* DOES NOT EXIST

INSTALLATION PARAMETER *z* MAY NOT BE ALTERED (where the address value of the installation parameter is outside the range of GEORGE's table of parameters)

PARAMETER NAME IS MISSING

VALUE OUT OF RANGE: MAX = *x* MIN = *y* (where the number parameter is greater than the maximum or less than the minimum allowed for this installation parameter)

z IS NOT A CORRECTLY FORMED NAME (parameter name)

JOB (JB)

Function

Introduces a job description to the system from a basic peripheral or MOP terminal, for running as a background job.

Format

JOB *job name, user name, terminator, MODE(mode), PARAM(a₁,a₂, . . . a_n)*

The *job name* must not coincide with the name of any of the user's files or with the name of any other job which the user currently has in the system. Only the first two parameters are mandatory, and, although they may be in either order, they must be the first two parameters of the command.

The *terminator*, **MODE** and **PARAM** parameters are optional and may be written in any order after the *job name* and *user name*.

If the *terminator* parameter is omitted, the standard terminator of four asterisks is assumed, and the terminator is stored at the end of the job description.

The **MODE** parameter specifies the type (**GRAPHIC**, **NORMAL** or **ALLCHAR**) of the basic file created to hold the job description; if no **MODE** parameter is given a **GRAPHIC** file is created.

The **PARAM** parameter specifies the user's job parameters, *a₁,a₂, . . . a_n*, which must conform to the rules that govern the formats of all parameters (see *Operating Systems GEORGE 3 and 4*, Chapter 10). They will be substituted in the appropriate places in the job description (denoted %A, %B etc.), when these places are encountered at the time the job's commands are being obeyed. They may also be used for accounting purposes.

Note that the previous method of specifying the parameters, that is:

JOB *job name, user name, terminator, a₁,a₂, . . . a_n*

will still be accepted in the current mark but it is recommended that users adopt the format detailed above.

Forbidden contexts

USER, OPERATOR

As a result of the command, the context of the new job becomes **USER** and **OFFLINE**. A **JOB** command may be issued only at command processor level zero.

Execution

GEORGE first checks that the user is in the index of permitted users and that he is allowed to start background jobs with the **JOB** command. When starting a job online, **HIGH** and **NORMAL** security users must type in their password, and **HIGH** security users are not permitted to start background jobs from a reader (see the **SECURITY** command). The job name is checked to see that it is a valid job name and that the user has no other files or jobs of this name. The command is abandoned if any of these requirements is not satisfied.

A monitoring file and a temporary working job description file are opened with the name given in the first parameter of the command. The job description is then read up to the terminator and filed in the working job description file, by means of an internally issued **INPUT** command.

The job is begun by the command issuing the file name (that is, the job name) as a macro command with the *a₁*, if any, as parameters.

Once the job is started, commands are read and obeyed from the working job description file until an **ENDJOB** command is encountered or the end of the file is reached. If an **ENDJOB** command is found, the job is terminated and the file is erased. If no **ENDJOB** command is found, the job is first suspended for a length of time determined by the value of the **WAITTIME** installation parameter (see the **INSTPARA** command); then, provided that it is not connected to a MOP terminal during the suspension period, the job is terminated and the file is erased.

Examples

JOB :JOHN,JOBA

JOBA is the job name. :JOHN is the user name. The terminator will be the standard terminator ****, which will be stored.

JOB OFFLINEJOB,:FRED,SENDJ

OFFLINEJOB is the name that will be given to the job. :FRED is the user name. The job description will be terminated by a line beginning ENDJ, which is to be stored.

JOB J3,:BILL,PARAM(INFILE,O,OUTFILE)

J3 is the name that will be given to the job. :BILL is the user name. The job description will be terminated by the standard terminator of ****, which will be stored. INFILE,O and OUTFILE are job parameters.

JOB A,:JAMES,PARAM(47,ABC),—
MODE(NORMAL),T####

A is the job name. :JAMES is the user name. The job description will be input into a NORMAL file (only relevant in the current mark for paper tape). The job description will be terminated by a line commencing ####, which will not be stored. 47 and ABC are job parameters.

Error messages

JOB NAME PARAMETER MISSING

JOB NAME *z* NOT UNIQUE

USER NAME PARAMETER MISSING

THE LIMIT ON THE NUMBER OF JOBS HAS BEEN REACHED

z HAS OVERDRAWN HIS MONEY BUDGET

z IS NOT A CORRECTLY FORMED NAME (job name or user name)

z IS NOT A REAL USER

z IS NOT A VALID JOB NAME

USER *z* IS EITHER PSEUDO OR ABOUT TO BE ERASED

USER *z* IS NOT ALLOWED TO START JOB UNLESS LOGGED IN

USER NAME/PASSWORD INVALID

THE SECURITY OF USER *z* DOES NOT ALLOW THIS COMMAND TO BE OBEYED IN THIS CONTEXT (no comment) (where the internally issued INPUT command is in error. This message will be preceded by the relevant INPUT error message.)

Logging messages

STARTED *user name job name date time* TYPE *job type*

This is the first message sent to the monitoring file of the background job.

LABEL (LB)

Function

Labels tapes which are not secure, including virgin tapes.

Format

LABEL *unit number, magnetic tape description*

The *unit number* is the geographical unit number of the deck on which the tape to be labelled will be loaded.

The *magnetic tape description* should be in entrant description format. It need not have an (*MT) qualifier but should contain the serial number of the tape to be labelled. If the tape local name is omitted the tape will be labelled POOLVTAPE, but will not be put in the pool, a separate NEW command being necessary for this purpose. The reel and generation numbers will be set to zero unless otherwise specified.

Forbidden contexts

NOT OPERATOR, REMOTE

Execution

The command should be typed on the console before the tape is loaded. After the command has been issued the message

UNIT *n* RESERVED

is displayed on the operator's console.

The tape should then be loaded and the deck engaged. The tape is checked to ensure that it does not contain a secure header label and is then labelled as specified in the command. The tape is then unloaded. No access must be made to the tape before it has been unloaded. If no tape is loaded after a deck has been reserved, the deck may be freed by an UNLOAD command. If a tape was already loaded but free when the command was given, it will be unloaded and the deck then reserved.

Users who require this command for 1971 Type 5 decks must use the Restore Time GIN macro RR71 or an error will result. This macro stops GEORGE from reading and checking any header label of a tape loaded on a 1971 deck before writing a new header. Thus, if a secure tape is loaded on a 1971 deck, GEORGE will write a new header label provided RR71 has been used.

Examples

LABEL 22, (360145)

The tape loaded on unit 22 will be given a serial number of 360145 and labelled POOLVTAPE.

LABEL 26, (20164,MAGT4(6))

The tape loaded on unit 26 will be given a serial number of 20164, labelled MAGT4 and given a file generation number of 6.

Error messages

UNIT *z* IS ALREADY ALLOCATED

UNIT *z* IS WRONGED

NOT ALLOWED TO USE UNIT *z*

UNIT *z* IS NOT ON THIS INSTALLATION

z IS NOT A VALID NUMBER FORMAT

SERIAL NUMBER MISSING

TAPE ON UNIT *z* IS SECURE AND MAY NOT BE LABELLED

TAPE ON UNIT *z* HAS FAILED AND HAS NOT BEEN LABELLED

LISTINCINDEX (LX)

Function

- 1 Lists :SYSTEM.INCINDEX on the line printer in an easily understood format
- 2 Lists any wronged tapes in :SYSTEM.INCINDEX

Format

- 1 LISTINCINDEX
- 2 LISTINCINDEX W

Forbidden contexts

NOT OPERATOR

Execution

A job is run under the user :OPERATORS, using :MACROS.LXJDF as a job description file. If format 1 is used, the job loads and enters :DUMPER.PROGRAM G3DR (the program used by the macro LISTDUMP). The program lists a table showing the bit position, octal value and meaning for both the status (state of the increment) and magstate (state of the magnetic tape) words. It also produces a listing of :SYSTEM.INCINDEX giving, for each increment, the status, date, time, size (in 512-word blocks) and number of files, followed by pairs of words for each tape that the increment is on: the first word of the pair is the tape serial number and the second is the magstate word for that tape.

If format 2 is used, the job loads and enters :LIB.PROGRAM XKHS. This facility produces a listing of any wronged tapes in :SYSTEM.INCINDEX.

Error messages

No error messages are generated by the job itself.

LISTSERIAL (LS)

Function

Lists the serial numbers and associated user names of magnetic tapes held in :SYSTEM.SERIAL in numerical order of serial numbers and in alphabetical order of user names.

Format

LISTSERIAL

Forbidden context

NOT OPERATOR

Execution

The program :LIB.PROGRAMVK66 is loaded and entered. It prints out unmerged blocks of 125 serial numbers and user names. Magnetic tapes marked as 'temporarily owned' or 'to be returned' are regarded as pool tapes. The LISTSERIAL macro is run under the user :OPERATORS and can only be issued in the NO USER context.

Error messages

ERROR IN MACRO

MODE (MO)

Function

Sets the mode of operation of the specified (switchable) deck or informs GEORGE that the mode has been switched on a deck that is only switchable by the operator.

Format

MODE *unit number, mode, T*

The parameters may appear in any order.

unit number is the geographical number of the tape unit in question. If omitted, the command applies to all switchable decks which do not contain tapes.

mode is the character M followed by the required setting, constructed additively as follows:

Parity:	odd = 4, even = 0
Gap-setting:	short = 8, long = 0
Recording density:	'800' = 32, '556' = 16, '200' = 0

Note that short gap is not allowed with 800 b.p.i. density.

T is optional; if included, it indicates that the change is temporary and the previous value will be reverted to as soon as the deck is released.

Execution

If the specified deck is currently unloaded and switchable, the command causes it to be switched and the appropriate records updated. If a tape is currently on the deck, its header will be re-read in the new mode, after switching. If the deck is only switchable by the operator, the settings will be updated and the header re-read if a tape is loaded.

In all cases, if the tape is in use, the command will fail.

Examples

MODE 32,M36,T

This command will temporarily change deck 32 to odd parity, long gap and 800 b.p.i.

MODE M36

This command will change all free switchable decks to odd parity, long gap and 800 b.p.i. Any otherwise applicable deck that does not have 800 b.p.i. will be set at 556 b.p.i.

Error messages

MODE MISSING

MODE *z* IS NOT AVAILABLE

NOT ALLOWED TO USE UNIT *z*

UNIT *z* HAS FAILED TO SWITCH

UNIT *z* IS ALREADY ALLOCATED

UNIT *z* IS NOT ON THIS INSTALLATION

UNIT *z* IS WRONGED

UNIT *z* NOT SWITCHABLE

z IS NOT A VALID NUMBER FORMAT

z IS NOT A VALID MODE

z IS NOT AN ONLINE MAGNETIC TAPE

MOP(MP)

Function

Changes the state of a multiplexer, uniplexer or communications processor from being available for system use to being available for use on-line by object programs only, or vice versa.

Format

- 1 MOP ON, *number*
- 2 MOP OFF, *number*, PM

Format (1) changes to system use and format (2) switches off system use and makes the device available for connection to an object program. The second parameter is the operator's number of a multiplexer, uniplexer, or communications processor. The third parameter in Format 2 is optional and is appropriate only if the second parameter refers to a communications processor. PM indicates that no further output is to be made to the processor, thereby allowing a clear post mortem of the communications processor to be taken.

Forbidden contexts

NOT OPERATOR, REMOTE

Execution

A check is made to ensure that the number given is a multiplexer, uniplexer or communications processor.

FORMAT 1.

If the specified device is marked free, GEORGE will allocate the device to GEORGE itself and go into the starting up sequence. If the specified device is not marked free, that is, it is on-line to an object program, then the MOP command is not obeyed.

FORMAT 2

If the specified device is marked allocated to GEORGE, then any lines being used for MOP will be closed down and any jobs running from those lines will be abandoned. Jobs in limbo (see *Limbo*, page 42.5) are not affected. A message explaining why the jobs have been abandoned will be sent to each line unless the device is a communications processor and the PM parameter is present. The multiplexer or uniplexer will be left in such a state that it can subsequently be ONLINED to an object program.

An attempt to switch to the current state will get the reply either

UNIT *n* IS ALREADY MOPPED ON

or

UNIT *n* IS ALREADY MOPPED OFF

When the MOP command has been completed the operator will be informed with the message:

UNIT *n* : MOP ON COMPLETED

or

UNIT *n* : MOP OFF COMPLETED

An attempt to do a further MOP command on the same device before this message has appeared will result in either

MOP ON IS TEMPORARILY NOT ALLOWED ON UNIT *n*

or

MOP OFF IS TEMPORARILY NOT ALLOWED ON UNIT *n*

If an attempt is made to issue a MOP OFF command for a physical multiplexer which has certain of its lines ATTACHED to a conceptual multiplexer that is ONLINE to a job, the MOP OFF command will not be implemented and the operator will be informed with the message:

MOP OFF IS TEMPORARILY NOT ALLOWED ON UNIT *n*

Error messages

UNIT *z* IS NOT ON THIS INSTALLATION

UNIT *z* IS NOT SUITABLE FOR SYSTEM USE

NO CONFIGURATION IS DEFINED FOR UNIT *z* IN THE INSTALLATION DESCRIPTION FILE

UNIT *z* IS NOT IN THE REQUIRED STATE

NEW (NE)

Function

Introduces one or more magnetic tapes to the system as work tapes, pool tapes or owned tapes. It may also be used to claim ownership of a tape currently at Query Status and thought by :SYSTEM.SERIAL to be a pool tape.

Format

NEW *tape description*₁, *tape description*₂, . . . , *tape description*_n

The magnetic tape description has a different format if it is issued in USER context rather than NO USER context:

1 NO USER context

(*tape serial number, code*) (*MT,PROPERTY *magnetic tape properties*)

The code is optional and if given must be either P for a pool tape or W for a work tape. If the code is omitted the tape will become a work tape.

2 USER context

(*tape serial number, magnetic tape name*) (*MT,PROPERTY *magnetic tape properties*)

The magnetic tape name is optional.

In both cases the *MT qualifier is mandatory and the PR *magnetic tape properties* qualifier is optional. If the latter is given the tape introduced will have the specified tape characteristics; other properties will not be taken into account.

Forbidden contexts

Format 1: NOT (central) OPERATOR

Format 2: NO USER, MOP

Execution

1 NO USER context

An entry for each of the specified tapes is created in :SYSTEM.SERIAL indicating that each tape is a pool tape or work tape. The tapes that are to become pool tapes must be loaded so that they may be relabelled POOLVTAPE.

2 USER context

If the tape is not currently known to the system then a directory entry is set up in the directory implied in the tape description (the new tape must then belong to the proper user). If a local name was not given in the parameter then a name of 12 zeros is put in the directory until the tape is next loaded, at which time the file name on the header label can be accessed and used to overwrite the 12 zeros. A corresponding entry is set up in :SYSTEM.SERIAL and the user's SPACMT budget is checked and updated. All traps are initially set open for the user.

If there is already a tape of this serial number known to the system and thought to be a pool tape but in query status, then the tape is loaded and its name checked against the name given in the magnetic tape description. If they are the same a directory entry is set up as described above. The entry for this tape in :SYSTEM.SERIAL and the user's SPACMT budget are updated and the query status is removed.

If SPACMT is exceeded the following message is output:

ERROR IN PARAMETER z IN NE : USER IS OVERDRAWN ON SPACMT BUDGET

If the user has no SPACMT budget the message is:

ERROR IN PARAMETER z IN NE : USER HAS NO SPACMT BUDGET

Examples

USER context

NEW (124250) (*MT),(36225,MYTAPE) (*MT),(36103) (*MT,PR PE)

NO USER context

NEW (360222) (*MT),(366221,W) (*MT),(366222,P) (*MT)

Error messages

SERIAL NUMBER ALREADY GIVEN IN THIS COMMAND

DEVICE TYPE QUALIFIER IMPERMISSIBLE

DEVICE TYPE QUALIFIER MISSING

SERIAL NUMBER MISSING

z IS A XENOTAPE AND MAY NOT BE REFERRED TO BY THIS COMMAND

SERIAL NUMBER ALREADY KNOWN

THE POOL/WORK TAPE CODE IS INCORRECT

TAPE IN USE

YOU DO NOT OWN z

MT z HAS BEEN WRONGED BY OPERATOR

MT z CORRECTLY IDENTIFIED BUT NOT AVAILABLE

MAGNETIC TAPE FAILED

TAPE IS AT QUERY STATUS BUT NAME GIVEN DOES NOT AGREE WITH HEADER LABEL

ILLEGAL COMBINATION OF PROPERTIES

PAGES (PA)

Function

Specifies the number of pages that require repeating on a line printer after a restart.

Format

PAGES *peripheral description*, NUMBER *number*

where *peripheral description* is as described on page 66, and the NUMBER parameter is the number of pages to be repeated on the line printer. The parameters may be in either order and the letter N is sufficient to identify the NUMBER parameter.

Forbidden contexts

NOT OPERATOR

Execution

If an error restart occurs while a file is being listed on the line printer the system request:

PLEASE GIVE PAGES FOR ERROR RESTART ON UNIT *peripheral description*

is output. The operator then uses the PAGES command.

The *peripheral description* is checked to ensure that there is an extant request for an error restart with that *peripheral description*. The NUMBER *number* is checked against the value of the installation parameter IPAGE at the time the LISTFILE command was issued. An error message will be output if the *number* is too large to indicate the maximum permitted value. If the number of pages listed is less than *number* the maximum possible number of pages will be repeated. If the value of IPAGE is 1 or the number of pages listed is 1 no system request will be output and one page will be repeated.

If CANTDO is typed instead of PAGES a default value of 1 page will be assumed.

Any error in the PAGES command causes the command to be ignored and any outstanding requests for pages will continue to be repeated until either a correct PAGES command or a CANTDO command has been issued for each request.

Example

PA U14,N3

The listing of the file on unit 14 will continue, when the peripheral is engaged, with the last 3 pages being repeated.

Error messages

NUMBER PARAMETER MISSING

THERE IS NO EXTANT REQUEST FOR A RESTART WITH THIS UNIT NUMBER

THE MAXIMUM NUMBER OF PAGES IS *z*

PMPRINT

Function

Produces GEORGE postmortem listings from a magnetic tape to an on-line line printer or to an off-line line printer channel for subsequent LISTFILEing.

Format

1 PMPRINT IDEN,OLLP,*MT*magnetic tape description*

2 PMPRINT D(*number*₁,*number*₂, . . . *number*_{*n*}),OLLP,*MT*magnetic tape description*

where each *number* specifies the position of one post-mortem in a series of postmortems held on a GEORGEPMTAPE. If the D parameter is omitted it is assumed that a listing of all such post-mortems is required.

All parameters are optional, and may appear in any order. Only one *magnetic tape description* may be included.

Forbidden contexts

NOT OPERATOR

Execution

The macro uses the GEORGE subject program #XK65 (see *Listing postmortem dumps from magnetic tape*, page 61) to print specified or all postmortems (Format 2) or the identities of all postmortems (Format 1) held on a magnetic tape.

If *MT*magnetic tape description* is included the tape specified will be used, provided that it is a GEORGEPMTAPE; if the parameter is omitted the first on-line magnetic tape called GEORGEPMTAPE will be used.

If the OLLP parameter is included printing will be direct to an on-line line printer; otherwise the listing will be sent to a file for subsequent LISTFILEing. The listing will be headed DOCUMENT PMPRINT.

Examples

PMPRINT	All the postmortem dumps on the first tape on-line called GEORGEPMTAPE will be LISTFILEd.
PMPRINT *MT(376403)	All the postmortem dumps on the magnetic tape 376403 are LISTFILEd.
PMPRINT IDEN,*MT(376607)	The identity of the postmortems dumped on magnetic tape 376607 will be LISTFILEd.
PMPRINT D(3)	The third postmortem dump on the first on-line magnetic tape called GEORGEPMTAPE will be LISTFILEd.
PMPRINT OLLP,*MT(306736)	Magnetic tape *306736 will be opened and all the postmortems on it listed to an on-line line printer.

Error messages

Error messages indicating exception conditions in #XK65 are listed under *Exception conditions*, page 62.

PROPERTY (PR)

Function

Introduces the name of a peripheral property to the system.

Format

PROPERTY *property name (qualifiers),CONSOLE console description*

The property name is a string of up to 12 character (letters, digits, hyphens and spaces). The character string must commence with a letter (that is, in local name format) and spaces are significant.

For an account of the use of property name qualifiers, see Chapter 3, page 29. The qualifiers, which may be specified in full or in shortened form, are PERMANENT (PERM), TEMPORARY (TEMP), INCLUSIVE (INCL) and EXCLUSIVE (EXCL).

A maximum of two qualifiers may be specified. Either PERM or TEMP may be given (TEMP is assumed unless PERM is given) and either INCL or EXCL (EXCL is assumed unless INCL is given).

If the PERMANENT qualifier is given, a command requesting the property while it is not ATTRIBUTED will generate a command error. This is because a permanent property will not be the sort of thing that an operator can attribute or cancel on demand.

If the TEMPORARY qualifier is given, a command requesting the property while it is not ATTRIBUTED will generate a request to the operator to ATTRIBUTE the property to a particular peripheral.

If the INCLUSIVE qualifier is given, a command requesting the property will use only units to which the property has been ATTRIBUTED but the unit may also be used by other commands.

If the EXCLUSIVE qualifier is given, a unit to which the property has been ATTRIBUTED may not be used by commands not requesting the property.

The CONSOLE parameter is optional, and if present causes the property name given to be classed as a *console property* and used as a *cluster name*. For a description of the setting up of clusters see Chapter 10 of the manual *GEORGE 3 and 4 Operation Management*. The CONSOLE parameter defines a cluster console and may have one of two forms:

- 1 If the cluster console is the central operator's console, the *console description* has the form: U0 or 0
- 2 If the cluster console is a remote 7023 console, the *console description* has the form of a remote peripheral description (see page 66) except that for 7023s on 7020 terminals connected via character buffering multiplexers and uniplexers it is not necessary to specify an interface number. Where no interface number is specified, the first 7023 on the terminal is taken as the cluster console.

Console properties are assumed to be PERMANENT and EXCLUSIVE; therefore PERMANENT/TEMPORARY or INCLUSIVE/EXCLUSIVE qualifiers are ignored in PROPERTY commands that include a CONSOLE parameter.

Forbidden contexts

NOT OPERATOR, REMOTE

Execution

If the property name is not known to the system and is not the special property CENTRAL, then it, together with its qualifier, is added to the list in :SYSTEM.PROPERTY. If the property is to be a CONSOLE property then details of the console are entered in :SYSTEM.IDF.

If the property name is known to the system and is not a console property, a command error is generated.

If the property name is known to the system but is a console property, the command is assumed to be defining a replacement cluster console. Details of the previous cluster console are therefore removed from :SYSTEM.IDF and the new console's details are inserted. Subsequently, the new console will be used for cluster messages.

Examples

PROPERTY GASBILL(TEMP,EXCL)

PR AST (PERMANENT,INCLUSIVE)

PROPERTY (PR)

PR STEVENAGE(PERM,EXCL),CONSOLE U30.L10.IAA

Error messages

PROPERTY NAME PARAMETER MISSING

DELIMITERS MUST BE PAIRED

THIS COMBINATION OF QUALIFIERS IS ILLEGAL

ONLY ONE SET OF QUALIFIERS MAY BE GIVEN

z IS NOT A CORRECTLY FORMED NAME

PROPERTY z EXISTS

PROPERTY NAME 'CENTRAL' EXISTS AND MAY NOT BE CANCELLED

CANCELLATION OF PROPERTY

A property name is removed from the system by the CANCEL PROPERTY COMMAND.

Format

CANCEL PROPERTY *property name*

The property name must have been declared by a previous PROPERTY command.

Forbidden contexts

NOT OPERATOR,REMOTE

Execution

The record of the property is removed from :SYSTEM.PROPERTY (and from :SYSTEM.IDF if it is a CONSOLE property). If it was ATTRIBUTED to or ASSOCIATED with any unit, then the attribution or association is removed. If the property name had been specified in an outstanding LISTFILE or ONLINE command, a command error is generated.

The following message is output to the operator's console:

PROPERTY z HAS BEEN CANCELLED

Example

CANCEL PROPERTY GASBILLS

Error messages

PROPERTY NAME PARAMETER MISSING

PROPERTY NAME z UNKNOWN

z IS NOT A CORRECTLY FORMED NAME

PROPERTY NAME 'CENTRAL' EXISTS AND MAY NOT BE CANCELLED

REDON (RO)

Function

Simulates a command issued by a specified peripheral or indicates to GEORGE that a special tape or disc is to be loaded or engaged.

Format

1 Basic peripherals

REDON *peripheral description, command*

The first parameter defines the peripheral. The second parameter is the command that is to be obeyed as if it had been issued from this peripheral.

2 Magnetic tape

REDON *peripheral description, magnetic tape description*

The first parameter defines the deck.

magnetic tape description has one of two forms:

(a) For Xenotapes

(*tape serial number, name(generation number/reel number)*)

where *tape serial number* must have an X appended to it. The other components of the description are used for checking and are optional.

(b) For non-standard tapes

name (generation number/reel number) (NONSTANDARD)

In this case *name* is the reference for the tape and must be included. The other details are optional.

3 Discs

REDON *peripheral description, serial number*

where *peripheral description* defines the device and *serial number* is a number of the form #nnnnnn, which specifies the serial number of the storage unit.

Forbidden contexts

NOT OPERATOR

Execution

1 Basic peripherals

REDON can be used to simulate commands issued from paper tape or card readers. Thus it can be used to identify unheaded paper tape or cards. For example:

REDON U27,DOCUMENTVFRED

will cause GEORGE to identify the paper tape (for example) on device 27 as the document called FRED.

RO U29.L10.IAA1,JB REMOTE,;MANAGER

will cause GEORGE to treat the card reader IAA1 as if a card bearing the JOB command defined in the REDON had been read.

If the peripheral specified is WRONGed a command error is generated (see below).

2 Magnetic tape

The REDON command for magnetic tape is used when loading a xenotape or a non-standard tape. These tapes should not be loaded in the normal way as the latter will be rejected as non-standard and the former will cause any secure tape with the same serial number to be put into query status. The REDON command works in the following way:

REDON (RO)

The REDON command must precede the loading of the non-standard tape or xenotape in order to warn GEORGE that special action is required for the tape to be loaded on the specified deck; any tape already loaded on this deck and not in use is first of all unloaded automatically by the command. When GEORGE is ready for the tape to be loaded then the message UNIT *n* RESERVED is output on the operator's console. The operator may now load the tape. If the tape is said to be a xenotape in the command then any name and details given in the tape description are checked against the tape's header label. GEORGE also checks that the tape is not a secure tape, and the tape is finally left positioned after the header label. If the tape is said to be non-standard then GEORGE checks that the tape is not secure and leaves it positioned at start of tape. If the tape does not have a write permit ring and one is required or vice versa then the tape is unloaded (currently without an explanation).

3 Discs

The command indicates that the specified unit is non-standard and may be accessed only by a Q or S trusted program using either mode #1600 or mode #1700. If the non-standard device is disengaged, the command does not terminate until the device is engaged.

Examples

REDON U32,(276212X)

REDON U31,NEWTAPE (NONSTANDARD)

REDON U43,#7302

Prepares for the loading of the Xenotape on deck 32.

Prepares for the loading of the non-standard tape, to be addressed as NEWTAPE, on deck 31.

Prepares for the loading of the non-standard storage unit number 7302 on unit 43.

Error messages

1 Basic peripherals

COMMAND PARAMETER MISSING

NOT ALLOWED TO USE UNIT *z*

THE OPERATOR'S NUMBER PARAMETER IS MISSING

UNIT *z* IS ALREADY ALLOCATED

UNIT *z* IS WRONGED

2 Magnetic tape

UNIT *z* IS ALREADY ALLOCATED

UNIT *z* IS WRONGED

UNIT *z* IS NOT ON THIS INSTALLATION

TAPE ON UNIT *z* HAS FAILED

z DOES NOT SPECIFY NON-STANDARD OR XENOTAPE

NON-STANDARD HEADER LABEL ON UNIT *z*

TAPE ON UNIT *z* IS SECURE

ENTRANT DESCRIPTION PARAMETER MISSING

LOCAL NAME NOT SPECIFIED

RESERVATION ON UNIT *z* CANCELLED

XENOTAPE ON UNIT *z* NOT CORRECTLY IDENTIFIED

XENOTAPES MUST NOT BE NONSTANDARD

3 Discs

REDON HAS BEEN OBEYED FOR UNIT *z*
A STANDARD SU IS LOADED ON UNIT *z*
NON-STANDARD SU *z* IS NOT AVAILABLE
SERIAL NUMBER *z* IS ALREADY ONLINE
SERIAL NUMBER PARAMETER MISSING
UNIT *z* IS NOT ON THIS INSTALLATION
SERIAL NUMBER IS INVALID

RETRIEVE (RV)

Function

Enables a file to be retrieved from a dump tape autonomously.

Format

RETRIEVE *file description*₁, *file description*₂, . . . , *file description*_n

Each parameter must uniquely specify the name of a file.

Forbidden contexts

None.

Execution

A retrieving activity, running independently of the user's job, will be set up for each file specified, unless the file is already on-line.

Error messages will be produced for command errors, such as an incorrectly formatted name, that can be detected immediately. If the file is already on-line, the following message will be sent in the COMMENT category:

z IS ALREADY ONLINE

If several files are already on-line, the message is as follows:

FILES ALREADY ONLINE:- z₁, z₂, z₃, ...

If the files are not on-line the user will not be able to determine the final status of the file (that is, whether or not it has been successfully retrieved) except by attempting to open it (by ASSIGN or LOAD commands, etc.) or by issuing another RETRIEVE or a CANCEL RETRIEVE command. Appropriate messages concerning the status of the file will then be produced.

When a job is set waiting for an implicit retrieve (for instance, when a LOAD command is issued for a file which is not on-line) the following message, indicating that retrieval has been started, is sent to the monitoring file:

z IS BEING RETRIEVED

If the user is at a MOP terminal and does not wish to wait, he can break in and continue to issue commands, such as QUIT; the file will still be retrieved.

The RETRIEVE command may be used in macros to initiate retrieval of files that will be needed during execution of the macro but may not be on-line when it is entered.

Example

RETRIEVE :LIB.PROGRAMVXE20, :LIB.SUBGROUPS-CM | Two activities will be set up to retrieve the files
| PROGRAMVXE20 and SUBGROUPS-CM

CANCELLATION OF RETRIEVALS

Function

Cancels a previously issued RETRIEVE command, or an implicit retrieve (see *Execution* below).

Format

CANCEL RETRIEVE *file description*

Forbidden contexts

None, but the command must be issued in the same context as the command that started the retrieval. In addition, if the context is USER, the command must be issued from the same job as the command that started the retrieval; however, in NO USER context retrieval may be cancelled from any source.

Execution

If the retrieval was initiated by a RETRIEVE command it can be terminated by a CANCEL command unless the file is now on-line.

If the retrieval was implicit (for instance, due to a LOAD command) break-in causes it to become autonomous and again a CANCEL command can be used to terminate the retrieval.

Error messages

If no retrieval initiated from this job for this file is outstanding when a CANCEL command is issued, one of the following messages is output:

z IS ALREADY ONLINE

z IS NOT ONLINE

RIGHT (RI)

Function

Makes a peripheral available for use by GEORGE.

Format

RIGHT *peripheral description 1, peripheral description 2, . . .*

where the parameters specify the local or remote peripherals to be made available for use.

Forbidden contexts

NOT OPERATOR

Execution

If this command is issued from a remote console, it may be used only to RIGHT a peripheral in its own cluster. If it is issued from the central console, it may be used to RIGHT any central or remote peripheral.

Provided that the command can be issued from the issuing console and that the peripheral exists, the message

UNIT *peripheral description* RIGHTED

is output on the operator's console from which the command was issued. GEORGE will mark the peripheral RIGHT, irrespective of whether it was previously marked WRONG (see the WRONG command).

Examples

RI 24,U25,IAA2,I62

RI U29.L10.IAA1,IAA6,U60,I73,I72

Error message

z NOT RECOGNISED

RUNJOB (RJ)

Function

Initiates a background job defined by a filed job description.

Format

RUNJOB *jobname,username,filename,PARAM(a₁,a₂,...a_n)*

The first 3 parameters are mandatory although *jobname* and *username* are interchangeable: *jobname* must not coincide with the local name given in the third parameter unless the local name is part of a file name with the user name specified as being other than the proper user.

The *filename* is the name of the job description file.

The parameters *a₁,a₂,...a_n* if present, are the job parameters which must conform to the rules that govern the formats of all parameters (see the manual *Operating Systems GEORGE 3 and 4*, Chapter 10).

Note that the previous method of specifying the job parameters, that is

RUNJOB *jobname,username,filename,a₁,a₂,...a_n*

will still be accepted but it is recommended that users adopt the format detailed above.

Execution

GEORGE checks the parameters of the command for errors and, if the checks are not satisfied, the command is abandoned. Otherwise, a monitoring file is opened with the name given in the first parameter, and the specified filename is issued as a macro with the *a₁,a₂,...a_n* if any, as parameters. Once the job has started, commands are read and obeyed from the job description file until either an **ENDJOB** command is encountered or the end of the file is reached at which point the job is terminated. If no **ENDJOB** command is encountered, the job is suspended for a length of time determined by the **WAITTIME** installation parameter (see *INSTPARA* command, page 93).

Examples

RUNJOB OLDJOBNO1,:FRED,
JOBFILE,PARAM(*CR0,*LP0,10,11)

OLDJOBNO1 is the job name;
:FRED is the user name;
JOBFILE is the name of the existing job description
file. The other parameters are job parameters.

Error messages

FILE NAME PARAMETER MISSING

JOB NAME PARAMETER MISSING

JOB NAME *z* NOT UNIQUE

THE LIMIT ON THE NUMBER OF JOBS HAS BEEN REACHED

THE SECURITY OF USER *z* DOES NOT ALLOW THIS COMMAND TO BE OBEYED IN THIS CONTEXT

USER NAME IS NOT IN DICTIONARY

USER NAME PARAMETER MISSING

USER NAME/PASSWORD INVALID

YOU CANNOT SPECIFY A WORKFILE NAME

z HAS OVERDRAWN HIS MONEY BUDGET

z IS NOT A CORRECTLY FORMED NAME (job name or user name)

z IS NOT A REAL USER

SPEAK (SK)

Function

Causes a message to be sent to a specified operator's console or consoles.

Format

- 1 SPEAK *,text*
- 2 SPEAK *console property name,text*
- 3 SPEAK *,text,B*

Forbidden contexts

NOT OPERATOR

Format 1 may not be issued from the central operator's console.

Format 2 may be issued from any operator's console.

Format 3 may only be issued from the central operator's console.

Execution

FORMAT 1

The specified text is output on the central operator's console.

FORMAT 2

The specified text is output on the specified operator's console (central or remote).

FORMAT 3

The specified text is output on all remote operators' consoles.

The text, as sent, is also output on the console which issues the command. If a message contains commas it must, of course, be enclosed by parentheses or quotes. A console may not SPEAK to itself.

Examples

SPEAK	CENTRAL,I'M SIGNING OFF NOW	either of these will cause the message <i>console property name</i> :- I'M SIGNING OFF NOW to be output on both the central operator's console and the originating console.
SPEAK	,I'M SIGNING OFF NOW	
SK	, "MOPPING OFF, 10 MINS",B	will cause the message CENTRAL:- MOPPING OFF, 10 MINS to be sent to all remote operator's consoles.
SK	READING, WE'VE JUST RECEIVED YOUR TAPE PUNCH OUTPUT	will cause an exchange of messages between two remote operators' consoles.
SK	BRACKNELL, (OK, WE'LL SEND SOMEONE OVER TO COLLECT IT)	

Error messages

YOU ARE THE NOMINATED CONSOLE FOR PROPERTY *z*
YOU MAY NOT BROADCAST FROM A REMOTE CONSOLE
z IS NOT A CONSOLE PROPERTY

STOPLIST (SL)

Function

Deletes all or some entries in the file :SYSTEM.OUTPUT of files waiting to be listed following a LISTFILE command. Note that STOPLIST is not provided to stop a listing which is in progress; TERMINATE may be used to do this.

Format

- 1 STOPLIST *output peripheral type, PROPERTY console property name*
- 2 STOPLIST *output peripheral type*
- 3 STOPLIST *user name, job name, output peripheral type, file name*
- 4 STOPLIST *user name, job name*

Forbidden contexts

NOT OPERATOR, REMOTE (Format 1)

NOT OPERATOR (Formats 2, 3 and 4)

Execution

1 FORMAT 1

All entries corresponding to files waiting to be listed on the specified peripheral type on the cluster specified by the console property name are deleted from the file :SYSTEM.OUTPUT. This format allows the operator to delete all the required entries in the case where the specified peripheral was available when a LISTFILE command was analysed but has become unavailable before the listing could be produced.

If the console property name is not specified, all LISTFILE entries for a peripheral of the specified type are deleted if the listing was intended for a central device.

2 FORMAT 2

All entries corresponding to files waiting to be listed on the specified peripheral type on the cluster the console of which was used to issue the STOPLIST command are deleted from the file :SYSTEM.OUTPUT.

Note that the result is undefined if the specified peripheral is still available and a file is in the process of being listed. Once a peripheral has been allocated by GEORGE for a listing, that listing is considered to be in progress and this command cannot be used to stop the listing.

3 FORMAT 3

The entry or entries in :SYSTEM.OUTPUT, corresponding to the specified file waiting to be listed following a LISTFILE command, are deleted. This format allows the operator to delete the required entry when a user asks that the listing of a file be abandoned. It is assumed that the user requested the listing of the file by a LISTFILE command.

4 All entries in :SYSTEM.OUTPUT that correspond to files waiting to be listed following a LISTFILE command issued by the specified job are deleted.

If the required file is actually being listed, or if the LISTFILE command is being implemented the following message is output on the operator's console:

REQUIRED FILE BEING LISTFILED ON UNIT *n*

where *n* is the operator's number of the device.

The file name specified in the STOPLIST command must be exactly the same as that given in the user's LISTFILE command.

If there are no files of the type specified the following message is output on the operator's console:

NO SUCH FILES

Note that if format 1 or 2 is used, and a peripheral has already been allocated for the listing this message may occur even if the listing does not appear to have started, for example if the peripheral concerned has not yet been engaged.

Error messages

z IS NOT A TERMINAL FILE

INVALID NUMBER OF PARAMETERS

NO PARAMETER

PROPERTY NAME z UNKNOWN

z IS NOT A CONSOLE PROPERTY

TAPERIGHT (TH)

Function

Specified tapes are freed from the effect of a TAPEWRONG command, issued by the operator.

Format

TAPERIGHT *tape serial number*₁, *tape serial number*₂, ... *tape serial number*_n

The *tape serial numbers* must be qualified by *MT.

Execution

The command may be issued by the operator or a user. If issued by the operator, any relevant 'tape wronged by operator' markers will be removed; if the user issues the command for tapes which he owns, then any 'tape wronged by user' markers will be removed. When both 'tape wronged by operator' and 'tape wronged by user' markers are unset, the tape is available for normal use.

All the tapes specified in the parameters will be dealt with, apart from those about which error messages are output.

Example:

TAPERIGHT (360245) (*MT), (360246) (*MT)

Error messages

DEVICE TYPE QUALIFIER IMPERMISSIBLE

DEVICE TYPE QUALIFIER MISSING

MAGNETIC TAPE DESCRIPTION FOR THIS COMMAND MUST CONSIST OF SERIAL NUMBER ONLY

SERIAL NUMBER ALREADY GIVEN IN THIS COMMAND

SERIAL NUMBER MISSING

THIS COMMAND IS ONLY ALLOWED IN OPERATOR OR USER CONTEXT

z IS A XENOTAPE AND MAY NOT BE REFERRED TO BY THIS COMMAND

z IS NOT KNOWN TO THE LIBRARIAN

TAPES (TP)

Function

Produces a listing of all or part of the contents of :SYSTEM.SERIAL as selected by the parameters.

Format

TAPES *basic output peripheral type, magnetic tape qualifier*

The *basic output peripheral type* is optional; if it is included it must be *LP or *CP; if it is omitted, the listing will be output on a line printer. The listing may not be produced on a paper tape punch.

The *magnetic tape qualifier* must be POOL, OWNED or WORK to indicate which tape categories are to be listed. If this parameter is omitted all the tape entries in :SYSTEM.SERIAL are listed.

Forbidden contexts

NOT OPERATOR, REMOTE

Execution

The command initiates a system job with the job name TAPES, under the user name :OPERATORS, which loads and enters a subject program. The program produces a listing of the tapes in :SYSTEM.SERIAL which satisfy the condition implied by the magnetic tape qualifier. The serial number of each tape is written to a workfile. The program also writes the user name if the tape is 'owned'. Otherwise it indicates that a tape is a pool or a work tape, unless a listing of only pool or only work tapes has been requested. 'Owned' worktapes are not listed under the owned category and in a listing of all tapes or work tapes, the serial number of an 'owned' worktape is qualified by the user name.

The workfile used by the program is listed using a LISTFILE command. As the TAPES command is implemented by running a job, a monitoring file is produced and listed on the line printer when the job is completed. Any errors in the execution of the command will appear in this listing, they are not sent to the operator's console.

Examples

TAPES *CP, OWNE

TAPES WORK, *LP

TAPES

A listing of owned tapes will be output on a card punch.

A listing of worktapes will be output on a line printer.

A listing of all tapes will be output on a line printer.

Error messages

ERROR IN DEVICE TYPE IN TAPES

ERROR IN MACRO TAPES – MACRO ABANDONED

MAGNETIC TAPE QUALIFIER NOT RECOGNIZED

will be sent to the monitoring file following the appropriate command error message or program event message, if any other error occurs.

TAPEWRONG (TW)

Function

Records specified tapes as not available for use until a TAPERIGHT command is given for these tapes.

Format

TAPEWRONG *tape serial number*₁, *tape serial number*₂, . . . , *tape serial number*_n

The *tape serial numbers* must be qualified by *MT.

Execution

The command may be issued by the operator or a user.

If a tape is TAPEWRONGed by the operator, it is marked 'tape wronged by operator'. If a user TAPEWRONGs a tape, it is marked 'tape wronged by user'.

If a tape specified by the parameters is a dump tape, it is recorded as no longer available for restoring and dumping.

The command has no effect on a tape that has already been TAPEWRONGed.

Once the current user has finished with the tape nobody can access it until all 'tape wronged' markers for the tape have been removed by TAPERIGHT commands.

All the tapes described by the parameters will be dealt with except those about which error messages are output. Any requests for a tape which has been TAPEWRONGed will be automatically CANTDone.

The effect of TAPEWRONG can be cancelled by the TAPERIGHT command.

Example

TAPEWRONG (360245) (*MT), (360247) (*MT)

Error messages

DEVICE TYPE QUALIFIER IMPERMISSIBLE

DEVICE TYPE QUALIFIER MISSING

MAGNETIC TAPE DESCRIPTION FOR THIS COMMAND MUST CONSIST OF SERIAL NUMBER ONLY

SERIAL NUMBER ALREADY GIVEN IN THIS COMMAND

SERIAL NUMBER MISSING

THIS COMMAND IS ONLY ALLOWED IN OPERATOR OR USER CONTEXTS

z IS A XENOTAPE AND MAY NOT BE REFERRED TO BY THIS COMMAND

z IS NOT KNOWN TO THE LIBRARIAN

TERMINATE (TE)

Function

Terminates the activity of a local or remote basic peripheral.

Format

TERMINATE *peripheral description*₁, *peripheral description*₂ . . . *peripheral description*_n

where each parameter specifies a peripheral whose activity is to be terminated.

Forbidden contexts

NOT OPERATOR

Execution

If the command is issued from a remote console, it may be used only for a basic peripheral in its own cluster. If it is issued from the central console it may be used for any local or remote basic peripheral.

Provided that the command can be issued from the issuing console, a check is made that the peripheral is of a basic type (CR, LP, TR, TP, CP). If these conditions are satisfied, the system request is cancelled: otherwise an error is signalled.

If the request is cancelled, subsequent action depends upon the nature of the peripheral activity. If the peripheral is being used by an object program the command has the same effect as the command

CANTDO ENGAGE UNIT *z*

where *z* is the peripheral description (see the CANTDO command) and the object program will be put into a peripheral-failed condition.

If the peripheral is being used by the system the system operation that is in progress is terminated at the point reached. For example, if an INPUT or JOB command is being implemented, the TERMINATE command will close the file that is being created. If a LISTFILE command is being obeyed, the TERMINATE command will remove the appropriate entry from the output queue. The message

UNIT *z* TERMINATED

will be output on the operator's console from which the command is issued.

Examples

TERMINATE 4

TERMINATE 4,U8,U29.L10.IAA2,I72

Error messages

PERIPHERAL *z* NOT ALLOWED IN THIS COMMAND

UNIT *z* NOT IN REQUIRED STATE

TRANSPORTS (TR)

Function

Outputs a list of the operator's numbers of magnetic tape decks that are available for the loading of magnetic tapes.

Format

- 1 TRANSPORTS MODE *number*
- 2 TRANSPORTS PROPERTY *peripheral property names*

In both cases the parameter is optional. If it is omitted then the numbers of any available tape decks are output; if a standard characteristic has been declared by means of the STAPE GIN macro then only decks with the standard characteristic will be listed. If **MODE *number*** is specified, then only the numbers of decks that can be switched to the specified mode are output. (For details of modes see the MODE command). If **PR *peripheral property name*** is specified then only the numbers of decks with the peripheral property name specified are output.

Forbidden contexts

NOT OPERATOR

Execution

A reply is output on the operator's console in the format

UNITS AVAILABLE ARE: *decimal integer decimal integer. . . .*

The decimal integers are the operator's numbers of the magnetic tape decks that are disengaged or engaged but not in use at the time the decks were examined. If there are no decks available, the reply

NO DECKS AVAILABLE

will be output

No attempt is made to switch any decks to the specified mode.

Examples

TRANSPORTS

Reply (to operator's console)

NO DECKS AVAILABLE

TRANSPORTS

Reply (to operator's console)

UNITS AVAILABLE ARE:

21 22

No decks are available.

The operator's numbers of all the available decks are listed.

Error messages

MODE *number* CONTAINS ILLEGAL QUALIFIER

UNLOAD (UL)

Function

- 1 (a) Frees a basic input peripheral which is in any of the following states:
 - (i) Loaded with a document headed by a DOCUMENT command.
 - (ii) Loaded with a job headed by a JOB command and waiting because JOBLIMIT has been reached.
 - (iii) Loaded with a job or data headed by a JOB or INPUT command and waiting for one of the following:
 - A file to be erased
 - A backing store jam to be cleared
 - A file to be retrieved
 - A file to become available
- (b) Rewinds the magnetic tape on a specified deck and frees the deck, provided that the tape is not on-line to a user or in use by the system.
- (c) Terminates a LABEL or REDON command for a magnetic tape deck.
- 2 Frees (as defined in 1(b) above) all tape decks, or all decks with a specified property or properties.

Format

- 1 UNLOAD *peripheral description*₁, *peripheral description*₂, ... *peripheral description*_n
where each parameter is the peripheral description of a basic input peripheral or magnetic tape deck.
- 2 UNLOAD *MT(PROPERTY *peripheral property names*)
The qualifier is optional.

Forbidden contexts

NOT OPERATOR

Execution

- 1 (a) Basic input peripherals
If a basic input peripheral is specified and the peripheral is not on-line to an object program, the peripheral is disengaged and the name of the document, job or data that is loaded on it is removed from the filestore.
In the event of the command terminating successfully the operator is informed by the message
UNIT *peripheral description* FREE
(b) Single magnetic tape decks
If a magnetic tape deck is specified, the tape on the deck is rewound, provided that it is not on-line to a user. All information concerning the association between the decks and the tape is erased and the deck is thus freed for the loading of another tape.
In the event of the command terminating successfully the operator is informed by the message
UNIT *peripheral description* FREE:-MT*tsn name (reel,generation)* FAILS *n* REPEATS *m*
or in the case of non-standard tapes
UNIT *peripheral description* FREE:-NONSTANDARD MT *name (reel,generation)* FAILS *n* REPEATS *m*
In either case if *n* is zero FAILS *n* will be omitted and if *m* is zero REPEATS *m* will be omitted.
- (c) Any LABEL or REDON command issued for the specified deck but not yet implemented is terminated.

UNLOAD (UL)

2 All magnetic tape decks

If the PR qualifier is omitted all engaged decks not currently in use are unloaded, as described under 1(b) above. If the qualifier is present and a tape characteristic property is specified, all engaged decks with the specified characteristic that are not currently in use are unloaded. Other properties are ignored.

One of the following messages will be output:

UNITS $a, b, c, \dots, n$ FREE

or

NO UNITS UNLOADABLE

For each tape unloaded for which a fail or repeat has occurred, the same message will be output as for a single tape (see 1(b) above).

Examples

UNLOAD 33

UL 172

Error messages

UNIT z : MT t sn IS ONLINE TO JOB NUMBER n

UNIT z : MT t sn IS IN USE BY THE SYSTEM

UNIT z IS NOT ENGAGED

UNIT z IS WRONGED

UNIT z IS NOT IN REQUIRED STATE

VIDEO

Function

Activates the Geronimo system (see Appendix 2)

Format

VIDEO CORETRACE,NOCHECK,REFER

All the parameters are optional and may be specified in any order. Any parameter included must be given in full; no abbreviations are allowed.

Context

The macro may be issued either from the operator's console or under user :MANAGER.

Execution

Assuming the macro to have been previously set up in the filestore, Geronimo is activated. If issued from the operator's console, the macro issues a RUNJOB command to create a job VDU—SYSTEM running under user :MANAGER.

The effect of each parameter included is as follows:

<i>Parameter</i>	<i>Effect</i>
CORETRACE	Suppresses on-line tracing into filestore (see <i>Postmortem and diagnostic tracing facilities</i> , page 175). Use of this parameter will have beneficial effects on Geronimo performance.
NOCHECK	No security checks will be carried out within the remote job entry vetting code. For further details see the section <i>Using the RJE facility</i> , page 173.
REFER	The system will re-instate the MASTER identity and context area prior to starting up the AVDU terminals (see <i>The MASTER AVDU</i> , page 171). This parameter causes the FINISH command to dump the above-mentioned details for subsequent restore when the program is next loaded with this parameter specified.

WHATJOBS (WJ)

Function

Outputs to the monitoring file (Format 1 only) or on the operator's console or a 7023 cluster console (Formats 1 or 2) either the number of current jobs or a list of all the current jobs and NO USER activities together with some information about each.

Format

1 WHATJOBS

2 WHATJOBS *level parameter*

where *level parameter* can be either any combination of NAME, URG, TIME or CORE, or ALL which is equivalent to specifying all the other parameters. The command may be issued from a MOP terminal in Format 1 only.

Forbidden contexts

Format 1 None

Format 2 NOT OPERATOR

Execution

FORMAT 1

Counts of the numbers of MOP and background jobs are output with the message:

n MOP JOBS, *m* BACKGROUND JOBS

If both counts are zero, the message:

NO USER JOBS CURRENTLY IN SYSTEM

is output.

Note that system issued jobs are not included in the count.

FORMAT 2

The output of the command consists of a line of headings followed by one line of information for each job or NO USER activity. If there are no such jobs, the following message is output:

NO JOBS FROM THIS CLUSTER

The headings and information output depend upon the parameters given and are as follows:

<i>Parameter</i>	<i>Heading</i>	<i>Information given</i>
Any	JOB NO	The number by which the system identifies the job (see the WHATSTATE command).
Any	MOP?	This field may contain information in the following formats: 1 Background job (central): N 2 Background job (remote): Rz where <i>z</i> is either: (a) For a job loaded from a terminal peripheral on a 7020 connected via a character buffering multiplexer or uniplexer, the terminal identifier of the 7020 to which the job's cluster console is connected. or (b) For a job loaded from a terminal peripheral on a 7020 connected via a message buffering system, the identifier of the cluster console of that peripheral.

WHATJOBS (WJ)

Parameter	Heading	Information given
		3 MOP job: either
		(a) CI (command issuer)
		or
		(b) either
		(i) Uxx.Lxx (multiplexer)
		or
		(ii) Uxx (uniplexer)
		(c) L if the job is in limbo (see <i>Limbo</i> , page 42.5)
NAME	USER NAME. JOB NAME	Either the user name and job name or NO USER as appropriate.
URG	UR	The urgency of the job (see Chapter 5, page 58).
URG	CP	The Computing Power Index which is output as a percentage (see Chapter 5, page 58).
TIME	JOB TIME	The time used so far by the job.
TIME	PROG TIME	The time remaining for the program.
CORE	CORE (GEORGE 3 only)	The number of words of core store required by the object program. If this is greater than or equal to 10K, the number is output as the next higher multiple of K (where K equals 1024 words).
CORE	QUOT (GEORGE 4 only)	The number of words in the program's quota.

If a job has no core image, the PROG TIME and CORE (or QUOT) fields, if specified in the parameters, are left blank.

If a job has been saved by the High Level Scheduler, SAVED is output in the URG field.

Note: See note 3 on page 155.

Error messages

PARAMETER FORMAT ERROR

WHATLIST (WL)

Function

Produces a listing of some or all of the files waiting to be listed, or in the process of being listed, following a LISTFILE command.

Forbidden contexts

NOT OPERATOR

Format

- 1 WHATLIST
- 2 WHATLIST *output peripheral type*
- 3 WHATLIST *job name, user name*

Execution

If no parameter is given, a line of information is output on the operator's console for each file being listed or waiting to be listed. If a peripheral is specified, information is provided about each file waiting to be output on that device, or currently being output on that device. If a job name and a user name are specified, a line of information is output for each file for which that job has initiated a LISTFILE (unless the LISTFILE has been completed).

The format of the output is:

user name. job name peripheral type file name

followed by a list of the PROPERTIES required for the listing of this file.

If the command is issued from the central console, any LISTFILES relevant to remote peripherals will have the letter R output with the peripheral type. If issued from a cluster console, only LISTFILES destined for the cluster are listed (in this case the letter R is not output in the peripheral type).

If there are no files of the type requested, the message

NO SUCH FILES

is output.

Error messages

z IS NOT A CORRECTLY FORMED NAME (job name or user name)

z IS NOT AN OUTPUT DEVICE

PARAMETER NULL

WHATPER (WP)

Function

Outputs on the operator's console information about the state of a specified peripheral, or of all peripherals of a specified type.

Format

- 1 WHATPER *peripheral description*
- 2 WHATPER *peripheral type*

Forbidden contexts

NOT OPERATOR, REMOTE

Execution

Information will be output about the current state of the peripheral specified (Format 1), or about all peripherals of the type specified (Format 2), as described below.

In Format 2, if *peripheral type* is specified as *DA the information output will be about all UDAS devices (EDS, FDS and EDS 30); if *ED is used EDS units only will be reported on.

Notes

- 1 In certain instances a peripheral will be in an intermediate state and appear to be under GEORGE's control, when the reply given will be one of those listed under Set 6 of *Remaining peripherals* below.
- 2 Because of the slow transfer rate of the console typewriter, information may no longer be accurate by the time that it is output.
- 3 When the joblimit is reached (see the INSTPARA command) if a further job is submitted to the system the peripheral reading the JOB command will stop reading. Reading of the command will be continued when another job has been completed.

If the peripheral concerned is disengaged it is possible to use it to read cards or tape, other than those containing a JOB command, in the normal way. Thus, for example, INPUT could proceed normally. If this procedure is carried out, the system will forget that a JOB command was read on that peripheral.

COMMUNICATIONS DEVICES

If the peripheral specified is a uniplexer, multiplexer or a 7900 communications processor I.P.B. (*CC type 24) the message output will be one of the following:

UNIT *n* ONLINE TO JOBNO *m*, :*username.jobname*

UNIT *n* HAS MOP SWITCHED ON

UNIT *n* HAS MOP SWITCHED OFF

UNIT *n* WRONGED

UDAS DEVICES

If the peripheral specified is a direct access device (*DA, *ED or *FD) the message will be one of the following:

UNIT *n* BEING USED BY SYSTEM

UNIT *n* ENGAGED

UNIT *n* ENGAGED (HOLD SET)

UNIT *n* IDLE

Any of the first three of the above messages will be followed by *storage unit information*, consisting of a list of the serial numbers of storage units (SUSN) on the specified hardware units; each SUSN is followed, if applicable, by a list of all files open on that unit, with their file generation numbers. The format is:

SUSN *number*

FILES OPEN ON THIS UNIT ARE:

FILE F.G.N.

*filename*₁ *number*

*filename*₂ *number*

.

.

.

*filename*_{*n*} *number*

The (HOLD SET) submessage is output if applicable.

EXOTIC PERIPHERALS

For a graph plotter (*GP, type 20), universal document transport (*UD, type 23), or visual display unit (*ID, type 14 or GD type 21) the message will be one of the following:

UNIT *n* ONLINE TO JOBNO *m*, :*username.jobname*

UNIT *n* IDLE

UNIT *n* WRONGED

INTER PROCESSOR BUFFER (*PB, TYPE 28)

The message will be one of the following:

UNIT *n* IN USE BY *m* LINKS

UNIT *n* IN USE BY *m* LINKS BUT INOPERABLE

UNIT *n* NOT IN USE BUT OPERABLE

UNIT *n* IDLE

DRUMS

If the peripheral is a type 9 drum (*DR) or type 25 high speed drum (*HD) the message will be one of the following:

UNIT *n* BEING USED BY SYSTEM

UNIT *n* ONLINE TO JOBNO *m*, :*username.jobname*

UNIT *n* IDLE

REMAINING PERIPHERALS

The message will be selected from one of the following sets:

SET 1

UNIT *n* ONLINE TO JOB NO.*n*, *user name.job name*

FAILED LAST TRANSFER (if applicable)

Magnetic tape information (see below)

SET 2

UNIT *n* IDLE

WHATPER (WP)

SET 3

UNIT *n* WRONGED

SET 4

UNIT *n* RESERVED BY SYSTEM BUT NOT ENGAGED

SET 5

UNIT *n* HAS DOCUMENT *document name* LOADED

or

UNIT *n* LOADED BUT NOT BEING USED

Magnetic tape information (see below)

SET 6

UNIT *n* BEING USED BY SYSTEM

or

UNIT *n* BEING USED BY SYSTEM (JOBLIMIT REACHED)

The (JOBLIMIT REACHED) submessage is output if applicable and concerns slow input peripherals only (see Note 3 above). The message may be followed by magnetic tape information (see below).

SET 7

UNIT *n* AWAITING AVAILABILITY OF FILE :*username.local name*

Occasionally, for example if the file waited for is a frozen file, the user name and local name are not output, and the message is:

UNIT *n* AWAITING AVAILABILITY OF FILE

MAGNETIC TAPE INFORMATION

Magnetic tape information consists of one or more of the following messages:

1	T.S.N. <i>number WPR</i>	{ (WORKTAPE IN USE BY: <i>username</i>) for a worktape in use by a job but not on-line to a program (WORKTAPE) for a worktape in any other state (POOL TAPE) (XENOTAPE)

2 INSECURE TAPE

or

(NONSTANDARD TAPE)

followed by

tape name, REEL *number*, FGN *number*, RET *number*

Note that the reel number, file generation number, and retention period will be output if any of the three is non-zero.

3 LIBRARIAN TAPE OWNED BY: *username* (ABOUT TO BE RETURNED TO THE POOL)

4 *n* USER(S) IS/ARE WAITING FOR THIS TAPE

5	TAPE WRONGED BY	{ USER OPERATOR

6 ABOUT TO BE WITHDRAWN FROM SYSTEM (following a DEAD command)

In some cases not all the relevant information will be available, and in this case the message:

NO FURTHER DETAILS AVAILABLE

will be output.

For slow output peripherals (Tape punch, line printer, card punch) and IPB (type 28) devices information is output about all the properties attributed to the particular device (if there are any properties) under the heading:

PROPERTY CATEGORY MODE

Error messages

PARAMETER MISSING .

NO PERIPHERALS OF TYPE *z* ON THIS INSTALLATION

EXPRESSION *z* INVALID

WHATPROPS (WA)

Function

Outputs to the operator's console, or 7023, or to a MOP console, a list of all the properties in the system. Also outputs whether the property is permanent or temporary, inclusive or exclusive and whether it is a 'CONSOLE' property.

Format

WHATPROPS

Forbidden contexts

This command must be issued in either OPERATOR or MOP context.

Execution

The output of the command consists of a line of headings followed by a line for each property. The headings are:

PROPERTY NAME EXCL? PERM? CONSOLE?

A 'Y' is output under the heading EXCL? if it is an EXCLUSIVE property, under PERM? if it is a PERMANENT property, and under CONSOLE? if it is a 'CONSOLE' property.

Example

PROPERTY NAME	EXCL?	PERM?	CONSOLE?
GASBILLS	Y		
FAST		Y	
STEVENAGE	Y	Y	Y

WHATSTATE (WS)

Function

Outputs to the operator's console a detailed description of the state of either a specified job or every job and NO USER context activity.

Format

- 1 WHATSTATE *jobname,username,level parameter*
- 2 WHATSTATE *job number,level parameter*
- 3 WHATSTATE *ALL,level parameter*

The job number is a number by which the system identifies a particular job that has been presented to it. The operator can find this number by means of the WHATJOBS command.

The *level parameter* is optional and may be one of the following:

- 1 WAIT
- 2 PROG(RAM)
- 3 CONT(EXTS)
- 4 ALL

More than one parameter may be specified. If ALL is specified, all of the other three parameters are assumed.

Only the first four characters of each parameter are significant.

Execution

If Format 1 or 2 is used, information is output about the specified job.

If Format 3 is used, information is output about every job and NO USER context activity.

If the command is issued from a cluster console, then only information concerning jobs initiated from that cluster will be given. If there are no such jobs, the following message is output:

NO JOBS FROM THIS CLUSTER

OUTPUT

The headings and information output depend on the parameters given. The headings and information given with the WHATJOBS command is referred to and can be found on page 120.

<i>Parameter</i>	<i>Headings and information given</i>
None	As for command WJ ALL
WAIT	As for command WJ NAME
PROG	As for command WJ ALL
CONT	As for command WJ NAME
ALL	As for command WJ ALL

If Format 3 is used, the heading line is output only once, at the start of the information.

Context information (CONT or ALL parameters given)

The heading CONTEXTS is output followed by a line giving the contexts applying to the job when it is examined.

Waiting information (WAIT, ALL or no parameters given)

If the job is waiting, the following information is output:

WAITING:— *awaited event*

Core image information

In Format 1 if the context is CORE IMAGE, and ALL or PROG is specified as a level parameter, information is output under the following headings:

- 1 (a) In GEORGE 3 the following headings:

FAIR-WAIT LAST-TIME-RUN

where FAIR-WAIT is the fair waiting time in milliseconds and LAST-TIME-RUN is the time at which the program was last run. If the program has not been run since it was LOADED, the letter L is output in this field.

- (b) In GEORGE 4 the following headings:

FAIR-WAIT LAST-TIME-RUN PAGETURNS SIZE

where FAIR-WAIT and LAST-TIME-RUN are as described in 1(a) (above). PAGETURNS represents the number of pageturns that the program has done to date (modulo 4M) and SIZE refers to the size of the program.

2 PERIPHERALS

followed by information about on-and off-line peripherals allocated to the program. The following details are given:

- (a) For on-line peripherals, the peripheral type and unit number followed by the operator's number.
 (b) For off-line peripherals, the peripheral type and unit number followed by the file name.

Notes

- 1 The information provided by this command may be supplemented or changed from time to time in the light of operating experience.
 2 A job under examination may be in a transitional state. In this case some information may be unavailable or, in certain rare circumstances, misleading. If a job is in a transitional state, one of the following lines may be printed:

BEING BROKEN IN ON
 FURTHER DETAILS NOT CURRENTLY AVAILABLE
 NO LONGER ANY CORE IMAGE
 JOB DELETED FROM SYSTEM
 JOB HAS NOW BEEN SAVED

If an item of information is not available, a question mark is printed in the appropriate position.

- 3 Until the output from a WHATJOBS, WHATLIST, WHATPER, WHATPROPS or WHATSTATE command is complete any subsequent such command will be held waiting. An excessively long queue of such waiting commands will slow down the system significantly.

Error messages

NO SUCH JOB AS z

NO SUCH JOB NUMBER AS z

z IS NOT A CORRECTLY FORMED NAME

PARAMETER FORMAT ERROR

TRACING AND REPORTING ARE SUCH THAT YOU WILL GET NO OUTPUT FROM THIS COMMAND

WRONG (WR)

Function

Makes a peripheral unavailable for use by GEORGE.

Format

WRONG *peripheral description*₁,*peripheral description*₂, . . . *peripheral description*_n

where the parameters specify the local or remote peripherals to be made unavailable.

Forbidden contexts

NOT OPERATOR

Execution

If this command is issued from a remote console, it may be used only to **WRONG** a peripheral in its own cluster. If it issued from the central console, it may be used to **WRONG** any central or remote peripheral of the correct type. It is not permissible to **WRONG** a direct access device, an operating console or a MOP console.

Provided that the command can be issued from the issuing console and that the peripheral exists and is not busy, it is marked 'WRONG' irrespective of whether it was previously marked 'RIGHT' (see the **RIGHT** command). The message

UNIT *peripheral description* **WRONGED**

is output on the operator's console from which the command was issued.

If the peripheral is busy, it is marked 'Waiting to be wronged' and the message

UNIT *peripheral description* **WAITING TO BE WRONGED**

is output on the operator's console from which the command was issued. The peripheral is marked 'WRONG' when it is freed.

Examples

WRONG 4,21,I74

WRONG IAB6,U29.L10.IAB5,U60.I73,U4

Error messages

UNIT *z* **CANNOT BE WRONGED**

z **NOT RECOGNISED**

Appendix 1 Operator messages

This appendix comprises an index of messages that can be output to the operator's console by the system, as distinct from messages output by jobs. For each message a reference is made to the section heading or headings under which it is documented or, alternatively, a brief explanation is given.

<i>Message</i>	<i>Reference or explanation</i>
A NAME RECORD WAS EXPECTED	<i>General restore routine</i> , page 44
A TRAPS RECORD WAS EXPECTED	<i>General restore routine</i> , page 44
ABOUT TO BE WITHDRAWN FROM THE SYSTEM	<i>WHATPER</i> , page 148
ABOVE THRESHOLD ON RESIDENCES $n_1 n_2 n_3 \dots$	<i>The backing store unjammer</i> , page 46
ALL BACKGROUND JOBS FINISHED: n MOP JOBS LEFT	<i>Method of closedown</i> , page 63 and <i>FINISH</i> , page 97
ALL USER JOBS NOW FINISHED	<i>Method of closedown</i> , page 63 and <i>FINISH</i> , page 97
AN INCOMPLETE DIRECTORY ENTRY ENCOUNTERED	<i>General restore routine</i> , page 44
ASSOCIATION NOW { EXISTS { NON-EXISTENT	<i>ASSOCIATE</i> , page 75
AWAITING REWIND: DECK n	<i>Action after postmortem complete</i> , page 61
BACKING STORE JAM CLEARED AT <i>time</i>	<i>The backing store unjammer</i> , page 46
BACKING STORE JAM:— DUMPER STARTED AT <i>time</i>	<i>The backing store unjammer</i> , page 46
BACKING STORE JAM:— UNREASONABLE LIMIT ON IP FORMULA — PLEASE ALTER TO INCLUDE VALUE m	<i>The backing store unjammer</i> , page 46
BACKING STORE UNJAMMER STARTED ON <i>date</i> AT <i>time</i> (GEORGE TIME <i>time</i>)	<i>The backing store unjammer</i> , page 46
BAD COMPILATION	<i>The GEORGE Loader</i> , page 12
BEING BROKEN IN ON	<i>WHATSTATE</i> , page 154
BLKS READ FROM MT NOT SAME AS IN SSF SENTINEL	<i>General restore routine</i> , page 44
BROKE IN <i>segment name</i> AT ADDRESS <i>number</i> OBEYING <i>instruction</i>	<i>GEORGE first error</i> , page 59
CAN THIS JOB'S LIMIT BE INCREASED?	<i>Control of on-line backing store</i> , page 22.4
CANCEL ALLOCATE FILES FROM RESIDENCES $n_1 n_2 n_3 \dots$	<i>The backing store unjammer</i> , page 46
CARTRIDGE *nnnnnn NOT FOUND IS IT AVAILABLE?	<i>Early Morning Start routine (E.M.S.)</i> page 22
CARTRIDGE/UNIT NOT FOUND	<i>Allocator/Loader directive error messages</i> , page 17
COMMAND FORMAT ERROR	<i>Allocator/Loader directive error messages</i> , page 17
COMMAND NOT RECOGNISED	<i>General restore routine</i> , page 44

Message

DELETED CC, FI, WA or WU

DELETED OK

DIALOGUE THROUGH IPB z $\left\{ \begin{array}{l} \text{SUSPENDED} \\ \text{RESUMED} \end{array} \right.$

DISC NOT NEWLY INITIALISED

DISPLAY LP UNIT X FIX

DISPLAY RP or RW

DO TAPERIGHTS PLEASE

DO YOU STILL WISH TO DUMP ON n MTS?

DRUM n CORRUPT – PLEASE INITIALISE

DRUM n FREE

DRUM n FREE: PLEASE ALLOCATE FILES

DUMP n NOT FOUND, DELETED OK

DUMP TAPE z IS ENGAGED

DUMP TAPE TSN, DECK NO?

DUMPER INTENDS TO DUMP ON MTS
list of tape serial numbers

DUMPER: MT *tape serial number* NOT USED DUE TO
FAILURE HISTORY

DUMPER: MT *tape serial number* NOT USED DUE TO
UNAVAILABILITY

DUMPER: MT *tape serial number*
 $\left\{ \begin{array}{l} \text{RETURNED} \\ \text{TO BE REUSED} \end{array} \right\}$ - INCREMENTS OBSOLETE

DUMPER: MT
tape serial number $\left\{ \begin{array}{l} \text{WRITE FAIL} \\ \text{END-OF-TAPE} \end{array} \right.$

DUMPER REQUIRES POOL TAPE

DURING RELOCATION OF FILES

END OF INCREMENT n ON MTS *tape serial numbers*
 $\left\{ \begin{array}{l} \text{COMPLETE} \\ \text{INCOMPLETE} \end{array} \right.$

END OF PM DUMP

ERROR $\left\{ \begin{array}{l} \text{B} \\ \text{C} \\ \text{J} \end{array} \right.$

ERROR(S) DETECTED – RELOADING IN 10 MINS

Reference or explanation

The GIN program, page 10

The GIN program, page 10 and *The Journal Accounting Program*, page 58.3

Operational aspects of IPC, page 42.6

Allocator/Loader directive error messages,
page 17

The GIN program, page 10

The GIN program, page 10

Dump checking, page 45

The incremental dumper, page 47

Examples of use of the property system,
page 30

Examples of use of the property system,
page 30

Examples of use of the property system,
page 30

*Listing postmortem dumps from magnetic
tape*, page 61

General restore routine, page 44

Postmortem dump to magnetic tape,
page 60

The incremental dumper, page 47

The incremental dumper, page 47

The incremental dumper, page 47

Dump tape releasing, page 49

The incremental dumper, page 47

The incremental dumper, page 47

Allocator/Loader directive error messages,
page 17

The incremental dumper, page 47

Changing the postmortem dump tape,
page 60

GELL, page 11

The Journal Accounting Program, page 58.3

Message

ERROR IN *command*: THIS COMMAND IS NOT ALLOWED
IN NO USER CONTEXT

ERROR NUMBER *n* DETECTED

ERROR: THE MULTILINE CHARACTER IS ILLEGAL
FROM AN OPERATOR'S CONSOLE

ESF OR SSF SENTINEL WAS EXPECTED

ESF SENTINEL WAS EXPECTED

EVERYTHING HAS FINISHED

EXPRESS COMMAND ACKNOWLEDGED

EXPRESS COMMAND FOR *jobname*,
username ACKNOWLEDGED

EXPRESS COMMAND FOR JOBNUMBER
x ACKNOWLEDGED

EXPRESS COMMAND IS NOT MEANINGFUL
FOR A SYSTEM ISSUED JOB

FAILED LAST TRANSFER

FAILURES ON UNIT **nnnnnn* THEREFORE
FILENAME *filename* NOT FOUND
IS IT AVAILABLE?

FAULT M

FILE *filename* OVERFLOWED DURING
INPUT/JOB ON UNIT *n*

FILE NOT FOUND

FILE REQUESTED TOO BIG

FILENAME *filename* NOT FOUND IS IT AVAILABLE?

FILENAME ALREADY EXISTS

FILES ALREADY ONLINE:— *z*₁, *z*₂, . . .

FILESTORE MUST BE MARK
n + OR — 1

FILESTORE RESIDENCES FULL

FORMAT ERROR IN INC. NO. *n* ON IDT *z*

FURTHER DETAILS NOT CURRENTLY AVAILABLE

GEORGE BSTYPE UNKNOWN

GEORGE FIXED CORE TOO LARGE FOR THIS EXEC
GEORGE READY

GEORGE 3 Mk *m.n* READY

Reference or explanation

Command errors, page 68

TTTP messages, page 54

GEORGE commands, page 65

General restore routine, page 44

General restore routine, page 44

Method of closedown, page 63 and
FINISH, page 97

EXPRESS, page 95

EXPRESS, page 95

EXPRESS, page 95

EXPRESS, page 95

WHATPER, page 148

Early Morning Start routine (E.M.S.),
page 22

GELL, page 11

INPUT, page 103

Allocator/Loader directive error messages,
page 17

Allocator/Loader directive error messages,
page 17

Early Morning Start routine (E.M.S.),
page 22

Allocator/Loader directive error messages,
page 17

RETRIEVE, page 129

*Hardware, other software, operator or
system limitation errors*, page 62.1

*Hardware, other software, operator or
system limitation errors*, page 62.1

General restore routine, page 44

WHATSTATE, page 154

The GEORGE Loader, page 12 and
Allocator/Loader directive error messages,
page 17

The GEORGE Loader, page 12

Early Morning Start routine (E.M.S.),
page 17

Operating clusters, page 42.2 and *Setting Up
the I.D.F.*, page 42

Message

HALTED:— CL, PC, PL, PO, PT, WI or WT

HALTED:— CP, CR, LP, TP or TR

HALTED:— NOT ENOUGH CORE AVAILABLE

HAVE ANY TAPES NOW BEEN MADE AVAILABLE?

HLS { EMERGENCY
SWITCHING

IF YOU GO ON INCREMENTS WILL BE LOST

ILLEGAL FILENAME ON STEERING LINE

ILLEGAL JAP JOBNAME/USERNAME

INCINDEX HAS BEEN RECONSTITUTED

INCORRECT INCREMENT DUMP NUMBER FORMAT

INCREASE COREOBJECT FOR JAP

INCREMENT?

INCREMENT NO *n* NOT ON TSN *z*

INCREMENT *n* ABANDONED AND OBLITERATED

INCREMENT *n* CANNOT BE PROCESSED BECAUSE
MTS *tape serial numbers* ARE UNAVAILABLE.
IF THE TAPE(S) WILL NEVER BE AVAILABLE,
INCREMENT WILL BE MARKED AS PROCESSED AND
COPIES OF FILES IT CONTAINS WILL BE LOST
:DUMPER,PROC QUESTION: WILL THE TAPES EVER
BE AVAILABLE?

INCREMENT *n* NOT PROCESSED BECAUSE IT IS
ON CURRENT DUMP TAPES

INCREMENT *n* NOT PROCESSED BECAUSE THERE
ARE NO LATER RELIABLE INCREMENTS ON
AVAILABLE TAPES

INCREMENT *n* ON MTS *tape serial numbers* OBSOLETE

INCREMENT *n* ON MTS *tape serial numbers* SUITABLE
FOR GENERAL RESTORE

INCREMENT *n* PROCESSED
DO YOU WANT TO CONTINUE PROCESSING?

INCREMENT(S) *list of increments* OBSOLETE ON MT(S)
list of tape serial numbers

INCREMENT(S) PROCESSED: *n* TO *m*

INSECURE TAPE

INSTALLATION PARAMETER
'*parameter name*' = decimal integer

INVALID CYL.NO(S)

INVALID FILENAME

INVALID MESSAGE

Reference or explanation

The GIN program, page 10

The loader program #XKOF, page 9
and page 21

The loader program #XKOF, page 9
and page 21

TTTP messages, page 54

The operation of the high level scheduler,
page 57

TTTP messages, page 54

The loader program #XKOF, page 9
and page 21

The Journal Accounting Program, page 58.3

TTTP messages, page 54

General restore routine, page 44

The Journal Accounting Program, page 58.3

Early morning start routine (E.M.S.)
page 22 , and *General restore routine*, page 44

General restore routine, page 44

The incremental dumper, page 47

The dump tape processor, page 50

The dump tape processor, page 50

The dump tape processor, page 50

The dump tape processor, page 50

The incremental dumper, page 47

The dump tape processor, page 50

DUMP, page 93

TTTP messages, page 54

WHATPER, page 148

INSTPARA, page 105

Allocator/Loader directive error messages,
page 17

Allocator/Loader directive error messages,
page 17

Postmortem dump to magnetic tape,
page 60

Message

INVALID SERIAL NO.

IS A CONFIGURATION UPDATE REQUIRED ON UNIT *n*

IS A TTTP POSTMORTEM REQUIRED?

JA JOB ALREADY EXISTS

JAM CLEARED ON RESIDENCES *n*₁ *n*₂ . . .

JAM ON RESIDENCES *n*₁ *n*₂ *n*₃ . . .

JAP SWITCHING

JAP TERMINATED – COMMAND ERROR

JAP TERMINATED – ERROR(S) DETECTED

JOB DELETED FROM SYSTEM

JOB HAS NOW BEEN SAVED

JOBLIMIT REACHED ON UNIT *peripheral description*

jobname, username, IS BEING ABANDONED

jobname, username, WILL BE ABANDONED AS SOON AS POSSIBLE

LAST SSF SENTINEL READ *filename*

LIBRARIAN TAPE OWNED BY: *username*
(ABOUT TO BE RETURNED) TO THE POOL

LIMIT INCREASED TO *n* KWORDS

LIMIT OF ONLINE BACKING STORE REACHED

LISTDUMP MUST BE RUN IN ORDER TO PROCEED

LISTFILE OF *filename* IN *username. jobname* HAS BEEN LOST AS DEVICE NO LONGER EXISTS IN CLUSTER *property*

LISTFILE OF *filename* IN *username. jobname* HAS BEEN LOST DUE TO LOST FILE *username. filename*

LISTFILE OF *filename* IN *username. jobname* HAS BEEN LOST DUE TO PERMANENT EXCLUSIVE PROPERTIES

LISTFILE OF *filename* IN *username. jobname* HAS BEEN LOST DUE TO STOPLIST

LOADED

LOAD TAPE *tape serial number*

LOST MY TAPE

MAGNETIC TAPE DECK BUSY

MAGNETIC TAPE GEORGEPM TAPE – FAIL, DELETED MT

MAGNETIC TAPE GEORGEPM TAPE FORMAT ERROR, DELETED ER

Reference or explanation

Allocator/Loader directive error messages, page 17

Early Morning Start routine (E.M.S.), page 22 and *Setting Up the I.D.F.*, page 42

TTTP messages, page 54

The Journal Accounting Program, page 58.3

The backing store unjammer, page 46

The backing store unjammer, page 46

The Journal Accounting Program, page 58.3

The Journal Accounting Program, page 58.3

The Journal Accounting Program, page 58.3

WHATSTATE, page 154

WHATSTATE, page 154

Remote job entry, page 42.3

ABANDON, page 71

ABANDON, page 71

General restore routine, page 44

WHATPER, page 148

Control of on-line backing store, page 22.4

Control of on-line backing store, page 22.4

TTTP messages, page 54

Output, page 26

Output, page 26

Output, page 26

Output when a STOPLIST command is obeyed

GELL, page 11, and *Allocating the filestore residences, swap residences and the GEORGE chapter residence using the Allocator/Loader*, page 13

Changing postmortem dump tape, page 60

Allocator/Loader directive error messages, page 17

Hardware, other software, operator or system limitation errors, page 62.1

Listing postmortem dumps from magnetic tape, page 61

Listing postmortem dumps from magnetic tape, page 61

Message

MOP {ON
OFF} IS TEMPORARILY NOT
ALLOWED ON UNIT *n*

MT CHECKSUM FAIL

MT FAIL

MT *tsn name (reel, generation)* FAILS *n* REPEATS *m*

MT FORMAT ERROR

MT *tape serial number*
IS {NOW
NO LONGER} SECURE

MT *n* KNOWN TO S/SERIAL BUT NOT TO S/INCINDEX

MTS *a, b, c*, ARE WRONGED, MAKING ALL FILES ON
INCREMENTS *d, e, f*, UNAVAILABLE
CAN YOU TAPERIGHT ANY OF THESE TAPES?

NAME IN SSF SENTINEL NOT NAME IN TOP FCB

NAME RECORD IN A DIRECTORY IS INCORRECT LENGTH

NEED LOGICAL FILE 3: CORRECT
FSDESC OR PUT FILE ONLINE

NEW INPUT TAPE NEEDED FOR PROCESSING

NO DECKS AVAILABLE

NO FILESTORE DESCRIPTION

NO ONLINE {HSD
1964 DRUM

NO FURTHER DETAILS AVAILABLE

NO {HLSEMERGENCY}
JOBSTARTER } EITHER STARTING AGAIN

NO INPUT TAPES AVAILABLE FOR INCREMENT

NO JOBS FROM THIS CLUSTER

NO LONGER ANY CORE IMAGE

NO LP AVAILABLE FOR POSTMORTEM

NO MORE CORE AVAILABLE – REALTIME ON?

NO OLD OUTPUT TAPES ONTO WHICH TO RESUME

NO OUTPUT TAPES AVAILABLE FOR PROCESSING

NO OUTPUT TAPES REQUIRED BY TTTP

NO POSTMORTEM DUMPS ON MT *tape serial number*,
DELETED OK

NO RECORD IN S/DUMPER OF COPY TAPE

NO RECORD IN S/INCINDEX OF INCREMENT *n*

NO RECORD IN S/INCINDEX OF MT *n*

NO RECORD IN S/SERIAL OF COPY TAPE

NO ROOM ON HSD

Reference or explanation

MOP, page 116

Allocator/Loader directive error messages,
page 17

Allocator/Loader directive error messages,
page 17

UNLOAD, page 141

Allocator/Loader directive error messages,
page 17

Pool tapes and work tapes, page 32

General restore routine, page 44

Dump checking, page 45

General restore routine, page 44

General restore routine, page 44

Hardware, other software, operator or system
limitation errors, page 62.1

TTTP messages, page 54

TRANSPORTS, page 140

Hardware, other software, operator or
system limitation errors, page 62.1

Allocator/Loader directive error messages,
page 17

WHATPER, page 148

The operation of the high level scheduler,
page 57

TTTP messages, page 54

WHATJOBS, page 145, and WHATSTATE,
page 154

WHATSTATE, page 154

Line printer output, page 60

TTTP messages, page 54

TTTP messages, page 54

TTTP messages, page 54

TTTP messages, page 54

Listing postmortem dumps from magnetic
tape, page 61

General restore routine, page 44

General restore routine, page 44

General restore routine, page 44

General restore routine, page 44

Allocator/Loader directive error messages,
page 17

Message

NO ROOM ON 1964 DRUM

NO SUCH FILES

NO SWAP RESIDENCE DECLARED

NO { :SYSTEM.HLSEMERGENCY }
{ :SYSTEM.JOBSTARTER }

CAN YOU PROVIDE?

NOT UDAS DISC

NO UNITS LOADABLE

NO USER JOBS CURRENTLY IN SYSTEM

NON STANDARD HEADER LABEL ON UNIT *n* label

NONSTANDARD MT name (reel, generation) FAILS *n*
REPEATS *m*

(NONSTANDARD TAPE)

OK

OK ONE SHOT FINISHED

PLEASE ATTRIBUTE PROPERTY *property names*
TO UNIT *peripheral description*

PLEASE CHECK *data description* ON UNIT *n*

PLEASE ENGAGE DECK HOLDING INCREMENT
DUMP NUMBER *n*

PLEASE ENGAGE UNIT *peripheral description*

PLEASE ENGAGE UNIT *peripheral description*
FOR ERROR RESTART

PLEASE FREE UNIT *peripheral description*
OF PROPERTY *property names*

PLEASE FREE UNIT *peripheral description*
OF PROPERTY *property names* AND
ATTRIBUTE PROPERTY *property names*

PLEASE GIVE PAGES FOR ERROR RESTART ON
UNIT *peripheral description*

PLEASE LINE UP UNIT *peripheral description*

PLEASE LOAD DOCUMENT *document name*

PLEASE LOAD WPR ON MT tape serial number
 UNLOAD
ON UNIT *number*

PLEASE LOAD { MT tape name
 MT tape serial number load condition
 POOL TAPE
 WORK TAPE

{ ON deck characteristic UNIT number number ... TSN *tsn₁* *tsn₂* Loading magnetic tapes, page 32
{ WHEN deck characteristic UNIT AVAILABLE

Reference or explanation

Allocator/Loader directive error messages,
page 17

STOPLIST, page 134, and WHATLIST,
page 147

Allocator/Loader directive error messages,
page 17

The operation of the high level scheduler,
page 57

Allocator/Loader directive error messages,
page 17

UNLOAD, page 141

WHATJOBS, page 145

Loading magnetic tapes, page 32

UNLOAD, page 141

WHATPER, page 148

GELL, page 11, and Restoring and loading
GEORGE using the Allocator/Loader,
page 16

The Journal Accounting Program, page 58.3

Property name qualifiers, page 29

Exotic peripherals, page 42.5

General restore routine, page 44

Engaging basic output peripherals, page 26
and INPUT, page 103

Failures during input, page 25, and
INPUT, page 103

Property name qualifiers, page 29

Property name qualifiers, page 29

Failures during output, page 27, and
PAGES, page 120

Output to special stationery, page 28

On-line data, page 24, and DOCUMENT,
page 92

Loading magnetic tapes, page 32

Message

PLEASE PUT STORAGE UNIT *nnnnnn* ONLINE

PLEASE RE-ALLOCATE

PLEASE RELOAD

PLEASE RELOAD RESIDENCE *n* OR ASK FOR A
GENERAL RESTORE

PLEASE SELECT { NONSTANDARD } *tape*
UNIT FOR { XENOTAPE } *description*
 { WPR }
 { NO WPR }
 MODE *n* FOR JOB
 NO *n*: *deck numbers*

POSTMORTEM REQUIRED?

PROCESSOR STARTING ON INCREMENT *n*

PLEASE LOAD MT *tsn* ON UNIT *m m m m*

PROCESSOR TERMINATED

PROGRAM OF JOB *jobnumber* SUCCESSFULLY SWAPPED
IN

PROGRAM OF JOB *jobnumber* TOO BIG FOR SWAPIN
UNDER PRESENT SYSTEM LOAD

PROPERTY *property name* HAS BEEN CANCELLED

PROPERTY *z* HAS BEEN CANCELLED

PUT LOGICAL FILE *n* ONLINE OR ASK FOR A
GENERAL RESTORE

READY

REALTIME PROGRAM(S) OF JOB(S) *jobnumber*₁,
*jobnumber*₂, . . . *jobnumber*_{*n*} PREVENTING SWAPIN BY
FRAGMENTING CORE

RECORD(S) DELETED FOR INCREMENT(S) *list of*
increments

RELOAD *number*

REPOSITIONING OF TAPE FAILED

REQUIRED FILE BEING IMPLEMENTED

REQUIRED FILE BEING IMPLEMENTED ON
UNIT *n*

REQUIRED FILE BEING LISTFILED ON UNIT *n*

RESIDENCE 3 DOES NOT CONTAIN A
FILESTORE

RESIDENCE 3 NOT DECLARED

RESIDENCE *m* REDUCED BY *nK* WORDS

RESIDENCES *n*₁ *n*₂ *n*₃ . . . CLEARED BELOW
THRESHOLD

RESIDENCES *n*₁ *n*₂ *n*₃ . . . NOW ABOVE
THRESHOLD

Reference or explanation

Loading direct access devices, page 36

Direct access devices, page 36

Allocator/Loader directive error messages,
page 17

Backmap, page 43

Loading magnetic tapes, page 32

GEORGE first error, page 59

The dump tape processor, page 50

The dump tape processor, page 50

Siting of object programs in core, page 22.4

Siting of object programs in core, page 22.4

Cancelling properties and attributions,
page 30

PROPERTY, page 123

Hardware, other software, operator or
system limitation errors, page 62.1

The GIN program, page 10 and page 22
and *Loading after system error*, page 62.1

Siting of object programs in core, page 22.4

DUMP, page 93

Paperlow, page 28

Allocator/Loader directive error messages,
page 17

CHANGELIST, page 87

CHANGELIST, page 87

STOPLIST, page 134

Hardware, other software, operator or
system limitation errors, page 62.1

Allocator/Loader directive error messages,
page 17

Early Morning Start routine (E.M.S.),
page 22

The backing store unjammer, page 46

The backing store unjammer, page 46

Message

RESTORE?

S/F *subfilename* IS NOT ON *filename*

SHALL I USE { :SYSTEM.HLSEMERGENCY?
 :SYSTEM.JOBSTARTER?

SIZE NEXT TAPE: OLD *n*; NEW *m*

SIZE SO FAR: OLD *n*; NEW *m*

SJFILE GEN NO LONGER EXISTS

SSF SENTINEL WAS EXPECTED

START OF DUMP NUMBER *n*

START OF INCREMENT *n* ON MTS *tape serial numbers*

subfile SUBFILE FORMAT IS INCORRECT

SYSTEM CAN NOW MEET OBJECTQUOTA

SYSTEM ERROR *n*

:SYSTEM.IDF HAS BEEN ADJUSTED TO MAKE IT
COMPATIBLE WITH :SYSTEM.PROPERTY
(AN IDFLIST IS ADVISED)

SYSTEM OVERLOAD: UNABLE TO SATISFY
OBJECTQUOTA BY *n*

:SYSTEM.PROPERTY RECONSTRUCTED—ONLY
PROPERTY 'CENTRAL' NOW EXISTS

TAPE FAIL ON MT*n*

TAPE FAILURE IN INC. NO *n* ON IDT *z*

TAPE ON DECK *nn* NON STANDARD HEADER LABEL

TAPE REPOSITIONED

TAPE WRONGED BY { USER
 OPERATOR

TAPE *tape serial number* INCORRECT HEADER LABEL

TAPE *tape serial number* LOAD WPR

THE LIST OF POSSIBLE TAPES IS AS FOLLOWS
tsn(W) tsn(W) tsn(W) . . .

THE SYSTEM HAS TEMPORARILY CLOSED DOWN

THIS COMMAND IS NOT ALLOWED IN *context name*
CONTEXT

THIS IS NOT A COMMAND

TOO MANY OUTPUT TAPES; LIST TRUNCATED

TRANSFER FAILURE ON LOADING; PLEASE RELOAD

Reference or explanation

Early Morning Start routine (E.M.S.)
page 22 , and *Backmap*, page 43

The loader program #XKOF,
page 9 and page 21

The operation of the high level scheduler,
page 57

TTTP messages, page 54

TTTP messages, page 54

The Journal Accounting Program, page 58.3

General restore routine, page 44

Postmortem dump to magnetic tape,
page 60

The incremental dumper, page 47

The loader program #XKOF, page 9
and page 21

Siting of object programs in core, page 22.4

GEORGE first error, page 59 and
GEORGE second error, page 62

Early Morning Start routine (E.M.S.)
page 22

Siting of object programs in core, page 22.4

Early Morning Start routine (E.M.S.)
page 22

The loader program #XKOF, page 9
and page 21

General restore routine, page 44

Changing postmortem dump tape,
page 60

Allocator/Loader directive error messages,
page 17

WHATPER, page 148

Changing postmortem dump tape, page 60

Changing postmortem dump tape, page 60

TTTP messages, page 54

GEORGE first error, page 59

Command errors, page 68

Command errors, page 68

TTTP messages, page 54

The GEORGE Loader, page 12 and
Allocator/Loader directive error messages,
page 17

Message

T.S.N. *number* WPR { (WORKTAPE)
(POOL TAPE)
(XENOTAPE)

TTTP IN QUIET MODE SO FORCED TO STOP

TTTP IS AWAITING A DUMP

TTTP NEEDS *n* POOL TAPE(S)

TTTPROCESSING UNDER OPERATOR CONTROL

TTTP TERMINATED – { COMMAND ERROR
ERRORS DETECTED
INCINDEX CORRUPTED?
INCORRECT PARAMETERS

TTTPROCESSING WITH ONLY *n* COPIES

UNABLE TO OPEN S/DUMPER

UNABLE TO OPEN S/INCINDEX

UNABLE TO OPEN S/SERIAL

UNIT *n* ATTACHED TO LINE *p* OF CONCEPTUAL *z*

UNIT *n* BEING USED BY SYSTEM: JOBLIMIT REACHED

UNIT *n* BUSY

UNIT *n* BUSY-OPAT

UNIT *n* DISENGAGED – PLEASE LOAD CARTRIDGE
z ON UNIT *n*

UNIT *n* FAIL

UNIT *n*:–FAIL

UNIT *n* FAIL *n*

UNIT *n* FAIL { TRACK **n* BLOCK **m*
WORD **n*
BLOCK **n*

UNIT *n*:– FAULT { N
P

UNIT *n*:– FIX

UNIT *number* FREE

UNIT FREE: MT *tsn* ALREADY ON UNIT *m*

UNIT *n* FREE: POOL/WORK TAPE NEEDS WPR

UNIT *n* HAS BECOME INOPERABLE

UNIT *n* HAS BEEN RECONSTRUCTED: PLEASE
REISSUE IDF COMMANDS

Reference or explanation

WHATPER, page 148

TTTP messages, page 54

TTTP messages, page 54

TTTP messages, page 54

TTTP messages, page 54

TTTP messages, page 54

TTTP messages, page 54

TTTP messages, page 54

General restore routine, page 44

General restore routine, page 44

General restore routine, page 44

ATTACH, page 78

WHATPER, page 148

May occur during output

Action after postmortem complete,
page 61

CARTRIDGE, page 86 and Exchanging
filestore cartridges, page 39

The GEORGE Loader, page 12 , and
Allocator/Loader directive error messages,
page 17

GELL, page 11

Changing postmortem dump tape, page 60

Direct access devices, page 36

GELL, page 11

GELL, page 11 , The GEORGE loader,
page 12 , Allocator/Loader directive error
messages, page 17 , Allocating the filestore
residences, swap residences and the
GEORGE chapter residence using the
Allocator/Loader, page 13 , and Direct
access devices, page 36

UNLOAD, page 141, Unloading magnetic
tapes, page 35 , and CANTDO, page 85

Loading magnetic tapes, page 32

Loading magnetic tapes, page 32

Inoperable MOP terminals and jobs in
limbo, page 42.5

Setting up the I.D.F., page 42

Messages

UNIT *n* HAS DOCUMENT *document name* LOADED

UNIT *n* HAS MOP SWITCHED $\begin{cases} \text{ON} \\ \text{OFF} \end{cases}$

UNIT *number* HEADER READ FAIL. RELOAD ON UNIT *number number* UNIT *number* FREE

UNIT *n* IDENTIFIER *m* HAS BECOME INOPERABLE

UNIT *n* IDLE

UNIT *n* IS ALREADY IN THE MOP $\begin{cases} \text{ON} \\ \text{OFF} \end{cases}$ STATE

UNIT *number* IS NOT ENGAGED

UNIT *n* IS WAITING FOR FILE *username, local name* TO BE FREE

UNIT *n* LINE *m* ATTACHED TO LINE *p* OF CONCEPTUAL *z*

UNIT *n* LINE *m* HAS BECOME INOPERABLE

UNIT *n* LOADED BUT NOT BEING USED

UNIT *n* :MOP OFF COMPLETED

UNIT *n* :MOP ON COMPLETED

UNIT *n* ONLINE TO JOB NO. *n, username.jobname*

UNIT *n* ONLINED

UNIT *n* RESERVED

UNIT *n* RESERVED BY SYSTEM BUT NOT ENGAGED

UNIT *peripheral description* RIGHTED

UNIT *n* TERMINATED

UNIT *peripheral description* WAITING TO BE WRONGED

UNIT *peripheral description* WRONGED

UNITS $z_1, z_2, z_3, \dots, z_n$

UNITS *n m* . . . INOPERABLE

UNITS AVAILABLE ARE *decimal integer₁ decimal integer₂ . . . decimal integer_n*

UNJAMMER STUCK:— PLEASE TRY ABANDONING JOBS (BUT NOT THOSE WAITING FOR BACKING STORE) OR RELOAD THE SYSTEM

UNLOAD *peripheral description*

username, jobname QUESTION: *text*

VERB FORMAT ERROR

VERB TOO LONG

Reference or explanation

WHATPER, page 148

WHATPER, page 148

Loading magnetic tapes, page 32

Inoperable MOP terminals and jobs in limbo, page 42.5

WHATPER, page 148

MOP, page 116

Unloading magnetic tapes, page 35

WHATPER, page 148

ATTACH, page 78

Inoperable MOP terminals and jobs in limbo, page 42.5

WHATPER, page 148

Early Morning Start routine (E.M.S.), page 22, Setting up the I.D.F., page 42 and MOP, page 116

MOP, page 116

WHATPER, page 148

An ONLINE command for a basic output peripheral has been satisfied.

LABEL, page 112, and REDON, page 125

WHATPER, page 148

RIGHT, page 131

TERMINATE, page 139

WRONG, page 156

Engaging basic output peripherals, page 26

WHATPER, page 148, CANTDO, page 85 and WRONG, page 156

UNLOAD, page 141

Allocator/Loader directive error messages, page 17

TRANSPORTS, page 140

The backing store unjammer, page 46

Unloading basic output peripherals, page 28

ANSWER, page 73

Command errors, page 68

Command errors, page 68

Message

VERSIONS OF GEORGE AND EXEC INCOMPATIBLE

WAITING BEFORE DICTIONARY UPDATE

WAITING:— *awaited event*

WHICH INCREMENT TO BE PROCESSED FIRST

WHICH INCREMENT TO STOP PROCESSING

WHILST REWRITING DIRECTORY

WRONG CONTEXT

WRONG DIRECTIVE ON STEERING LINE

z IS ALREADY ONLINE:— *z*₁, *z*₂, . . .

z IS BEING RETRIEVED

n IS NOT ON DECK *z*

n MOP JOBS, *m* BACKGROUND JOBS

z NOT INCREMENT DUMP TAPE

z NOT SUITABLE FOR RESTORING FILESTORE

n USERS ARE WAITING FOR THIS TAPE

timenow unitno FREE, *n* TRANSFERS, *m*
REPEATS, *l* FAILS

unitno, *n* TRANSFERS, *m* REPEATS, *l* FAILS

Reference or explanation

The GEORGE loader, page 12 and
Allocator/Loader directive error messages,
page 17

The Journal Accounting Program, page 58.3

WHATSTATE, page 154

TTTP messages, page 54

TTTP messages, page 54

TTTP messages, page 54

Allocator/Loader directive error messages,
page 17

The loader program #XKOF
page 9 and page 21

RETRIEVE, page 129

RETRIEVE, page 129

Changing the postmortem dump tape,
page 60

WHATJOBS, page 145

General restore routine, page 44

General restore routine, page 44

General restore routine, page 44

Output when a job releases an on-line
peripheral

Output at an interval specified by the
REPORTTIME installation parameter

Appendix 2 Geronimo

The following description of Geronimo is divided into four main sections. The division enables users of the system, managers or operators, to access only as much information as they require.

The first section is of general interest, and is an introduction to the system and its facilities. The details of setting up the system are in the second section and are intended for the manager. The third section describes the actual operating of the system, and the final section, of interest to any user, deals with errors and diagnostics.

INTRODUCTION

The Geronimo system runs under GEORGE 3 or 4 and provides the operator with requested information on one or more local Mark II alphanumeric visual display units (AVDU). No conceptual facility is provided however, therefore the whole of the cluster controller must be given over to Geronimo.

The system is very easy to operate and provides the following facilities:

- 1 It enables the operator to interrogate GEORGE, via an AVDU, about the present state of hardware, jobs and users in the system and to list files on the AVDU screen.
- 2 If requested by the operator, specified data will be automatically refreshed on an AVDU screen at pre-set intervals.
- 3 Short jobs may be entered and initiated by the operator from an AVDU.

The system is designed primarily to provide the operator with relevant information quickly and in an easily understandable form, but it has the additional benefits of reducing the traffic to and from the operator's console (through having information presented on another device) and of reducing the number of operator-initiated jobs processed by GEORGE.

Since Geronimo runs under the operating system as a normal job, it may be turned on or off at the operator's discretion, and any Geronimo system failures will not cause GEORGE to fail.

The system also provides postmortem and trace facilities, and a facility for Remote Job Entry, all of which are described in later sections.

IMPLEMENTATION

The Geronimo system is a macro and subject program running under GEORGE 3 or 4, and therefore in no way jeopardises the functioning of the overall operating system. No modifications to GEORGE are required apart from the trapping of the appropriate system files to the user under which Geronimo is to run.

System set-up

The system macro that runs Geronimo, together with the embedded subject program, is held in card form in a sub-file called VIDEO on the ICLKG3SYSTEM tape. The macro must be copied into a file called VIDEO under the user :MACROS using the COPYIN macro, and this file must be trapped to :MANAGER in READ and EXECUTE modes.

The following system files may be accessed by the VIDEO system to obtain information for various commands.

<i>File</i>	<i>Command</i>
:MANAGER.MDCOPY	RJE
:SYSTEM.OUTPUT	WLIST
:SYSTEM.JOBLIST	JOB
:SYSTEM.DOCUMENT	DECK
:SYSTEM.PARAMINDEX	PAR, TRACE (GEORGE 3 only)

<i>File</i>	<i>Command</i>
:SYSTEM.PARAMINDEXG4	PAR, TRACE (GEORGE 4 only)
:SYSTEM.SERIAL	POOL
:SYSTEM.INCINDEX	INDEX
:SYSTEM.PROPERTY	PROPERTY

It is necessary therefore to give :MANAGER a READ trap to each of these files before the system can be run. Failure to do so will result in the message:

VIDEO SYSTEM FAIL

being displayed on the operator's console, and a postmortem of the VIDEO system being taken.

Note: Since the RJE command requires access to :MANAGER.MDCOPY the system must be run under :MANAGER if this command is to be used.

USING GERONIMO

Activation

Geronimo is activated by a call to the VIDEO command macro from the operator's console. Details of the options that may be specified when this macro is called are given in the specification of the VIDEO command (see Chapter 8.)

It is not necessary to have all possible AVDU terminals active when the system is initiated since the system will cater for terminals being turned on or off, and take the appropriate action.

Basic rules for using Geronimo

- 1 When the system is first loaded, or restarted under the restart option, an introductory message is output to all active AVDUs. The user must then do a null input, that is, key in INPUT followed by SEND.
- 2 After the null input, the system will output the message COMMAND PLEASE. Any available Geronimo command (see *Normal commands*, page 171) may then be input. A command will only be accepted when the system has previously output COMMAND PLEASE.
- 3 Visible representation of each command input will begin where the cursor is stationed after the COMMAND PLEASE message.
- 4 All commands must be terminated with the SEND key being pressed, which is the signal for transmission.
- 5 If a command requires parameters, then these must follow immediately after the command and be separated from the command and from each other by commas. The SEND key must then be pressed in order to transmit the input. Trailing spaces are non-significant.
- 6 When an error message or a screenful of data is output, the user can get back to the 'COMMAND PLEASE' state by doing a null input as mentioned previously.

Note: In the case of commands that are longer than four characters, the first four characters will be sufficient to identify that command. Input to Geronimo may be in upper or lower case.

AUTO facility

The AUTO parameter may follow any of the following commands: DECK, JOB, STATE, WLIST. Its effect is to run the specified activity at predetermined intervals.

The period between refreshing the data for these commands varies according to the expected validity period of that data. The approximate intervals are as follows:

DECK	20 seconds
JOB	60 seconds
STATE	15 seconds
WLIST	20 seconds

If a command is issued with the AUTO parameter, the AVDU from which the command was issued is no longer considered to be in question and answer mode but under the complete control of Geronimo. Any subsequent keyboard depression that causes a transfer to be made, or a transfer aborted from Geronimo, signals Geronimo to send out the appropriate end of AUTO message and abandon the AUTO process. That AVDU can then be returned to question and answer mode by doing a null input. The system will then output the normal COMMAND PLEASE display.

Normal commands

The following is a list (in alphabetical order) of the commands available on any AVDU in the system.

<i>Command</i>	<i>Format</i>	<i>Function</i>
DECK	DECK DECK, AUTO	Gives details of on-line tape decks and serial numbers and headers of tapes on those decks.
HELP	HELP	Gives details of available system commands and a brief description.
INDEX	INDEX	Gives details of oldest known dump, last one processed, last one TTTPed and most recent restorable dump.
JOB	JOB JOB, AUTO	Gives details of user jobs currently running in the system, (jobnumber, username and jobname).
LIST	LIST, <i>filename</i>	Facility to list a specified terminal file on the AVDU screen.
Notes:		
1 Only the first 80 characters, excluding red tape data, of each record are output.		
2 Illegal characters are replaced by a letter N.		
3 'BREAK' input at the top of the screen will prematurely close the activity.		
4 A null input should be performed to get the next screen of the specified file.		
5 The specified file must be trapped so that the AVDU system's username can read it.		
MASTER	MASTER	Instates this AVDU terminal as 'MASTER' if identity not yet allocated. (See <i>The MASTER AVDU</i> , below).
PAR	PAR	Gives names and current values of set installation parameters.
POOL	POOL	Gives serial numbers of pooltapes and worktapes, differentiated by a P or W after the number.
PROPERTY	PROPERTY	Gives details of all properties currently known to the system.
RJE	(See <i>Remote Job Entry</i> , page 172).	(Remote Job Entry facility) Enables a user to submit a short job via the AVDU terminal.
STATE	STATE, <i>jobnumber</i> STATE, AUTO, <i>jobnumber</i>	Gives some details of a specified job such as CORE, milltime etc.
TRACE	TRACE	Gives the current values of the tracing installation parameters assembled in English.
WLIST	WLIST WLIST, AUTO	Gives details of those files which do not require properties that are waiting for output to central line printers.

The MASTER AVDU

One AVDU terminal out of the total active may acquire the status of MASTER. This status permits that AVDU to access additionally a set of MASTER commands which can be used to change the current system AVDU configuration. Initially, a user will have to identify himself as MASTER using the MASTER command described earlier. Subsequently, the installation management can choose to re-instate the MASTER identity, using the parameter mentioned in the VIDEO macro specification, or to have the MASTER identity vacant each time the system is loaded.

The MASTER AVDU can restrict the access of an AVDU to specified Geronimo commands by issuing a FORBID command (see *MASTER commands*, below). The MASTER AVDU cannot, however, restrict its own access to any

commands. Details of which commands are forbidden for use on which AVDUs are held in an area known as the *context area*.

Should the MASTER AVDU fail, a display is issued to the operator's console and the identity becomes vacant.

MASTER COMMANDS

In the following set of MASTER commands, n_1 is the group number and n_2 is the terminal address within that group.

<i>Command</i>	<i>Format</i>	<i>Function</i>
CLOSE	CLOSE, n_1, n_2	Closes the specified AVDU terminal. The specified AVDU terminal will be closed down on completion of the next READ or WRITE order, or if under AUTO, when that activity reaches its refresh time.
FINISH	FINISH	Closes the entire AVDU system down. If the REFER option is on (see <i>VIDEO command</i> , Chapter 8) then part of the closedown code dumps out the MASTER identity, and the context area (see <i>The MASTER AVDU</i> , page 171).
FORBID	FORBID, $n_1, n_2, command$	Forbids the use of a specified command by a specified terminal.
LINES	LINES	Gives the status values of all active AVDUs (see <i>Status values</i> , below).
NEWMASTER	NEWMASTER, n_1, n_2	Changes the MASTER AVDU's identity from the one issuing the command, to the one whose identity is given in the parameters.
OPEN	OPEN, n_1, n_2	Opens an AVDU terminal previously CLOSED either by command or owing to terminal failure.
PERMIT	PERMIT, $n_1, n_2, command$	Opposite to FORBID.

Status values

<i>Value</i>	<i>Significance</i>
1	Operator command in queue
2	Next read causes COMMAND PLEASE
3	Reading for a command
4	Terminal running under AUTO
5	Terminal running under LIST
6	Waiting to be closed
7	RJE data expected next
8	MESSAGE CLASH on last transfer

Close down

Since the Geronimo system has no special status when in action, it may be abandoned from the operator's console. However this method should only be used in emergencies.

Built into the program's code are gradual (that is, for specific AVDUs) and all-embracing close down commands. Both types of command can be issued only from the MASTER AVDU. Details of the CLOSE and FINISH commands are given in the previous section *MASTER commands*.

Remote Job Entry

The RJE facility allows the AVDU to be used as an input/output device for submitting short jobs to GEORGE and subsequently for receiving a log of what has been transmitted to and from GEORGE.

For a job to be submitted to GEORGE via this facility it must be possible to input the entire job source at one time on the AVDU screen. If the job is too long to fit on the screen at one time then it may not be submitted in this way. When the job is transmitted to Geronimo, the system opens an anticipated command issuer channel and submits the text to GEORGE. When this transfer is complete, or if GEORGE terminates the job for some error condition, the log is transmitted back to the user.

USING THE RJE FACILITY

Having received the COMMAND PLEASE invitation from Geronimo, the user inputs, for example:

RJE,username,password

where *username* is the username under which the job to be subsequently input is expected to run, and *password* is the password of this user, as if he were going to LOGIN to the system.

Geronimo will then carry out two checks, the first being that the supplied username does exist, and the second that the password was in fact the password of the specified user. If both username and password are correct, Geronimo outputs directions about screen layout.

If the NOCHECK option is operating, (see *VIDEO command*, Chapter 8) there are no checks carried out on username or password, and the user has simply to input RJE in response to COMMAND PLEASE in order to reach the stage where Geronimo outputs the direction on screen layout.

If any errors are detected (the username or password wrong or missing) then a suitable error message is output and the user can get back to the COMMAND PLEASE state by doing a null input.

For example, assuming that the initial RJE command call was correct, the system has output the directions for screen layout, and the user has erased the contents of the screen and input his entire job source on the AVDU screen, it would look like this:

```
JOB  TEST4, :MATT,T????
MYPASSWORD
EDIT  SOURCE(1)
T/LABEL1/,R/LABEL1/LABEL2/,T1
T/LABEL3/,P3,E
QPLAN  SOURCE(2)
SV  DUMP
AS  *CR,DATA
AS  *LP,LPFILE
EN  0
IF  CORE,DELETE
LF  LPFILE,*LP,NU
EJ  ALL
????
```

The user then presses the SEND key, whereupon Geronimo will perform the following checks, providing the NOCHECK option is not operating:

- 1 That the first line begins either JOB or JB.
- 2 That the username specified is the same as that supplied when the RJE command was input.

A suitable error message is output if either test fails. If the NOCHECK option is operating, the facility can be used for doing small file inputs and issuing NO USER context macros.

Assuming the job source to be syntactically correct, Geronimo will then transmit the assembled job to GEORGE via an anticipated command issuer channel. Geronimo keeps a log of transfers required for this activity and transmits this back to the user when the job has been issued, or if the command issuer channel becomes inoperable.

All user inputs to GEORGE (those input to the AVDU system) are displaced four characters to the right for clarity. This ensures that the user is both aware of any errors flagged by GEORGE and that he is informed of the job entering the host system.

When the log has been transmitted back to him, the user may return to the COMMAND PLEASE state by doing a null input. The characters], ↑ and \$ will be transmitted to GEORGE as part of the job submission phase, but will be replaced by space characters when the RJE log is transmitted back to the user.

As an example, the log for the job input example given above may look like this:

GEORGE 4 MK7 ON 23 OCT 72 AT 13.20.33

13.20.34

```
JOB TEST4, :MATT,T????
MYPASSWORD
EDIT SOURCE(1)
T/LABEL1/,R/LABEL1/LABEL2/,T1
T/LABEL3/,P3,E
QPLAN SOURCE(2)
SV DUMP
AS *CR,DATA
AS *LP,LFILE
EN 0
IF CORE,DELETE
LF LFILE,*LP,NU
EJ ALL
????
```

13.20.52

JOB SUBMISSION FINISHED AT 13.20.57

ERRORS AND DIAGNOSTICS

System-detected errors

If Geronimo detects a serious logic error, the subject program halts giving an identifiable code. (A list of the codes and their meanings is given here for reference.) The macro will take the appropriate postmortem action and then ask the operator if a restart is to be attempted.

<i>Code</i>	<i>Reason</i>
BR	Backing store read fail.
BW	Backing store write fail.
CI	Either a READ PERI to the CI channel during the RJE transmission has been rejected inoperable (that is, the CI job has been abandoned due to external intervention) or a command error has occurred in the commands issued to initiate the system. (A command error is normally due either to there not being enough REALTIME budget available or to it being impossible to on-line the AVDU controller).
DL	Data has been lost whilst reading the record from PARAMINDEX for the PAR command, or possibly for the TRACE command. (This is an unlikely error, since the code is prepared for a transfer of a maximum of 400 words.)
EB	EBM and 15AM not available.
F1	Unable to find the specified overlay in the overlay reference table.
F2	Backing store transfer fail from system overlay file.
G1	No vacant work zones currently available (effectively a work zone jam). There is no immediate cure except system restart.
L?	Attempt made to use a standard message generator for the LIST command which should build its own message in a private work zone.

<i>Code</i>	<i>Reason</i>
LC	Unable to match up status, having found a successful AVDU READ on a system AVDU.
LE	An attempt has been made to do a non-standard link for a standard process.
LS	Only slot available to receive overlay from the system backing store has the locked bit set (effectively an overlay slot jam).
M?	Either an attempt has been made to re-LINK for activity 15, which never requires this process (activity 15 runs the MASTER command), or an attempt has been made to use a standard message generator for the MASTER command, which is a direct process function.
NR	Not enough store allowed to set up activity blocks for all AVDUs found on-line. (Normally six).
O1	The operator's coordinator cannot match up the completed (issued) queue entry with any of the in-core blocks.
Q?	Read found on a status 1 (one with command in queue) but unable to find the queue entry.
R?	After the PM has finished, this halt allows the operator or macro to request that a restart should be attempted.
RF	Irrecoverable read fail from AVDU network.

GEORGE 3 and 4 detected errors

Command errors generated by commands issued directly from the Geronimo subject program are catered for by the subject program.

Command errors caused by errors in the macro job description cause the macro to abandon itself so that an analysis may be made of the resulting monitoring file.

Illegal program instructions are treated by the macro as if a programmed halt had occurred; that is, a postmortem is taken and a restart, if required, is attempted.

AVDU failure

Should an AVDU fail whilst under the control of Geronimo, a display is issued to the operator's console and that terminal is removed from the program's list of active AVDUs. Any files or command issuer channels open to that AVDU are released and that AVDU can be made operable again only by use of the OPEN command. Details are given in the section *MASTER commands*, page 172.

Major fails of the entire AVDU network cause the subject program to come to a specified halt. The macro will then produce a postmortem for subsequent analysis to determine the precise nature of the failure.

Postmortem and diagnostic tracing facilities

When there is a condition that requires a postmortem of the subject program, the macro does a normal core print via the standard GEORGE facilities and then enters the program's own postmortem procedure to print out tables, pointers and buffers in an easily readable form.

The subject program has own-code tracing routines which maintain the last 64 entrants in a cyclic buffer held in FIXED core. Optionally, the same code can also output to a terminal filestore file as it traces through the logic of the program. Regardless of whether on-line tracing was suppressed or not at macro call (see *VIDEO command*, Chapter 8), the postmortem procedure still prints out the contents of the trace buffer in the appropriate section of the postmortem.

Index

1901A remote job entry terminals	40	ATTRIBUTE	30, 81
7007/2 character buffering device	40	CONCEPTUAL	89
7020 remote data terminals	40, 42.2	FINISH	64, 98
7023 teletypewriter	40, 42.2	PROPERTY	30, 124
7071 teletypewriter	40	RETRIEVE	129
7083/1 termiprinter	40	Cancelling	
7084/1 termiprinter	40	attributions	30, 81
7085/1 termiprinter	40	characters on the console	5
7086/1 termiprinter	40	dumps	49
7900 Series processors	41	properties	30, 124
7920 character buffering device	40	CANTDO (CD) command	25, 27, 33, 42.4
7930 character buffering device	40		49, 85, 87, 103
ABANDON (AB) command	42.4, 71	CARTRIDGE command	39, 86
Accounting	3, 58.3	Cartridges	
AE—see ASSOCIATE command		exchanging	39
AH—see ATTACH command		CC—see CANCEL command	
Allocator/Loader	13, 16	Central job	42.4
directives	13, 16	CG—see CARTRIDGE command	
directive error messages	16	CHANGELIST command	42.4, 87
Alphanumeric visual display unit	169	CHAPTERQUOTA installation parameter	22.4, 105
failure	175	Chapter residence, GEORGE	7
MASTER	171	Character buffering devices	
ANSWER (AN) command	73	7007/2	40
ASSOCIATE (AE) command	42.2, 75	7920	40
ATTACH (AH) command	42, 42.2, 78	7930	40
ATTRIBUTE (AU) command	28, 30, 81	lines	40
Attributing properties	30	Checking,dump	45
AU—see ATTRIBUTE command		CL—see CONCEPTUAL command	
AUTO parameter to Geronimo commands	170	CLOSE Geronimo MASTER command	172
AVDU—see Alphanumeric visual display unit		Closing down the system	63
BACKGROUND installation parameter	58, 105	Cluster	
Background jobs, saving	58	definition	42.2
Backing store		job	42.3
online	22.4	operating	42.2
unjammer	46	Commands	
BACKJAM installation parameter	46, 105	contexts	68
Backmap	43	errors	68
system errors	43	format	65
Back-up system	43	operator, in alphabetical order	71 to 156
Basic peripherals	24	parameter formats	65
engaging	26	remote operator, available to	42.4
input to	24	Communications devices	39
output from	26	Computing Power Index (C.P.I.)	58
Breakdowns, system	59, 62.1	CONCEPTUAL (CL) command	42.1, 88
BROADCAST (BC) command	42.2, 83	Conceptual multiplexers	42.1
BSINTERVAL installation parameter	105	setting up	42.1
Budgeting	3	Console	
CANCEL (CC) command	30, 42.5, 84	nominated	42.2
use with commands:		properties	30
ASSOCIATE	76	CONSOLE parameter to PROPERTY	
ATTACH	42.5, 79	command	30, 76, 123
		CONTEXT installation parameter	105
		Control area involved in the failing	
		PERI order	37

COREOBJECT installation parameter	105	GEORGE loader	12
C.P.I.—see Computing Power Index		GIN	10
CPIRATIO installation parameter	105	#XKOF	9, 21
CRUNCHTIME GIN macro	59	Error restart on card reader	25
Data		Exchanging filestore cartridges	39
off-line	24	EXCLUSIVE property name qualifier	29, 123
on-line	24	Executive	
DATE installation parameter	105	loading GEORGE	11, 13
DATINTERVAL installation parameter	105	operator communication with	3
DCP—see Dedicated Control Program		Exotic peripherals	42.5
DEAD (DD) command	32, 42.4, 90	EXPRESS (EP) command	58.1, 96
DECK Geronimo command	171	Failures	
Dedicated Control Program	40, 41	input	25
DEFAULTBS installation parameter	22.4, 105	output	27
DG—see DISENGAGE command		File retrieval system	49
Direct access devices	36	Filestore	1
failure messages	37	cartridges, exchanging	39
loading	36	invalid	43
unloading	36	residences	7, 22
DISENGAGE (DG) command	65, 91	FINISH (FN) command	63, 97
DOCUMENT (DOCU or DM) command	65, 92	FINISH Geronimo MASTER command	172
Document	24	First error	59
<i>document name</i> parameter	92	FORBID Geronimo MASTER command	172
DUMP		Forbidden command contexts	68
command specification	93	FORMULA installation parameter	105
PROCESS	50	GELL (George Loader Loader)	11, 13, 62.1
RETURN	49	error messages	12
REUSE	49	General restore routine	44
TOTAL	49	GEO3DUMPTAPE	47
Dumper, incremental	47	GEO3PROCTAPE	51
Dumps		GEORGE	
cancelling	49	chapter residence	7
checking	45	commands	65
Dump tape		command parameter formats	65
changing postmortem	60	error messages	5
processor	50	Executive, loading the	11, 13
releasing	49	first error	59
DUMPTAPES installation parameter	47, 105	loading error messages	12, 16
DUMPTIME installation parameter	47, 105	Mean Time	22
Early Morning Start routine (E.M.S.)	22	mending	21
Editing	2	operator communication with	4, 65
E.M.S.—see Early Morning Start		reloading	59
Engaging basic output peripherals	26	restoring	9, 16
Engineer, maintenance	30	second error	62
Entrant properties	36	setting up with GEORGE Executive	13
EP—see EXPRESS command		setting up with operator's Executive	9
Errors		GEORGE PMTAPE	60
command	68	Geronimo	169
context	69	activation	170
first	59	AUTO parameter	170
hardware	62.1	closedown	172
in parameters	69	commands	171
in verbs	69	diagnostics	174
operator	62.1	errors	174
second	62	implementation	169
system	59, 62.1	MASTER AVDU	171
Error messages	68	MASTER commands	172
Allocator/Loader directive	16	postmortem	175
GELL	11	Remote Job Entry	172
GEORGE	5	set-up	169
		status values	172

GIN directives	12	JOBTRACE installation parameter	105
GIN program	10, 22	<i>jobname</i> parameter format	66
error messages	10	Journal Accounting Program	58.3
GMT—see GEORGE Mean Time		Juggernaut restorer	45
HARDWARE (HW) command	42, 100	L installation parameter	105
Hardware error	62.1	LABEL (LB) command	31, 112
Header label, magnetic tape	33	Legible headings, paper tape output from LISTFILE	27
Heading formats from LISTFILE	27	Librarian tapes	
Headings, legible paper tape from LISTFILE	27	definition	30
HELP Geronimo command	171	see also Secure tapes	
High level scheduler		Limbo	42.5
operation of	57	LINES Geronimo MASTER command	172
with FINISH command	63	LIST Geronimo command	171
HLSREVIEW installation parameter	58, 105	LISTDUMP system macro	52
HOLD button, to unload a U.D.A.S. device	36	LISTFILE command	26, 27, 29, 42.4
HOLDUP installation parameter	105	output heading formats from	27
HW—see HARDWARE command		LISTINCINDEX (LX) command	113
 		Listing postmortem dumps from magnetic tapes	61
ICLKG3MARK <i>n</i> tape	7	LISTSERIAL (LS) command	32, 114
ICLKG4MARK <i>n</i> tape	7	Loader program, #XKOF	9, 21
I.D.F.—see Installation Description File		Loading	
IDFLIST system macro	22.2, 42.1, 102	after system error	62
IDIV installation parameter	105	direct access devices	36
IMULT installation parameter	105	GEORGE	11, 16
INCLUSIVE property qualifier	29, 123	GEORGE Executive	11
Incremental dumper	47	magnetic tapes	32
INDEX Geronimo command	171	<i>local file name</i> parameter format	66
Informative output	4	Log analysis	58.3
INPUT (IN) command	23, 24, 65, 103	Low level scheduler	57
Input		LS—see LISTSERIAL command	
failures during	25	LX—see LISTINCINDEX command	
to basic peripherals	24		
Insecure tapes	31	:MACROS.EMSJOBS	45
Installation Description File (I.D.F.)	42	Magnetic tapes	31 to 36
altering during running	42.1	entrant properties	36
Installation parameters	22.3, 105	insecure	31
INSTPARA (IP) command	105	librarian (secure)	31, 32
Inter operator communication	42.5	loading	32
Inter Processor Buffer	42.6	non-standard	31
Inter Program Communication	42.6	peripheral properties	36
operational aspects of	42.6	secure	32
property system	42.7	standard	31
IP—see INSTPARA command		owned	32
IPAGE installation parameter	27, 105	unloading	35
IPB—see Inter Processor Buffer		virgin	31
IPC—see Inter Program Communication		<i>magnetic tape description</i> parameter format	66
 		Maintenance routines	46
Jam, backing store	46	:MANAGER.JUGGERNAUT	45
JOB (JB) command	23, 65, 110	MASTER Geronimo command	171
JOB Geronimo command	171	Mending GEORGE	21
Job	23	Message buffering multiplexers	40
central	42.4	identifiers	40
cluster	42.3	MESSAGETIME installation parameter	28, 105
description	1, 24	MINTRACE installation parameter	105
in limbo	42.5	MODE (MO) command	34, 115
MOP	23	Monitoring file	26
JOBLIMIT installation parameter	24, 42.4, 105	MOP (MP) command	42, 116
JOBTIME installation parameter	105	MOP (Multiple On-line Programming)	
		definition	2
		inoperable terminals	42, 5

job	23	off-line	24
jobs in limbo	42.5	on-line	23
with clusters	42.2	properties	36
MOPCPI installation parameter	105	remote bulk	42.3
MOPLIMIT installation parameter	24, 105	unusable	36
MOPTRACE installation parameter	105	<i>peripheral description</i> parameter format	66
Multiplexers		<i>peripheral property</i> parameter format	67
conceptual	42.1	<i>peripheral type</i> parameter format	66
message buffering	40	PERMANENT property qualifier	29
Multifiles	27	PMPRINT macro	61, 121
		POOL Geronimo command	171
NEW (NE) command	32, 34, 42.4, 118	Pool tapes	32
NEWMASER Geronimo MASTER command	172	Postmortem	59 to 62
Nominated console	42.2	action after	61
Non-standard tapes	31	dump to magnetic tape	60
		dump tape changing	60
		dumps, listing from magnetic tape	61
Object programs		Privacy	2
siting of in core	22.4	Processing the workload	22.3
OBJECTQUOTA installation parameter	22.4, 105	PROGTIME installation parameter	105
Off-line		Properties	29
data	24	attributing	30
peripherals	24	console	30
Off-lining, definition	2	entrant	36
OFR directive to #XKOF	9, 21	examples of the use of	30
On-line		on magnetic tapes	35
backing store	22.4	peripheral	29, 36
data	24	PROPERTY (PR) command	29, 123
peripherals	23	PROPERTY Geronimo command	171
OPEN Geronimo MASTER command	172	<i>property name</i> parameter format	67
Operating		Property name qualifiers	29, 123
clusters	42.2		
communication with Executive	4	Query status, security	34
communication with GEORGE	4, 65		
function of operator	3, 22.3	REDON (RO) command	32, 33, 42.4, 125
peripheral handling	23	RELOAD installation parameter	62, 62.1, 105
remote consoles	42.3	Reloading GEORGE	59
Operating systems, definition	1	Remote	
Output		batch processing	2
failure during	27	data terminal (7020 and 1901A)	40, 42.2
informative	4	Job Entry	42.3, 172
to monitoring file	26	RENAMEDIR command	34
to special stationery	28	Requests, system	4, 157
Owned magnetic tapes	32	Rerouting LISTFILES	42.4
		RES directive to #XKOF	9, 21
PAGES (PA) command	27, 42.4, 120	Residences	
PAPERLOW condition	28	filestore	7, 22
PAR Geronimo command	171	GEORGE chapter	7
Parameter		swap	7
formats	65	Resilience	2
qualifiers	68	Restart on card reader, error	25
<i>parameter name</i> parameter format	66	Restore	
Password system	2	pack/tape	9
PERMIT Geronimo MASTER command	172	routine, general	44
Peripheral		Restorer, juggernaut	45
basic	24	Restoring GEORGE	9, 16
central	42.2	RETRIEVE (RV) command	42.4, 129
clusters	42.2	RI—see RIGHT command	
exotic	42.5	RIGHT (RI) command	42.4, 131
handling	23	RJ—see RUNJOB	
identification	41	RJE Geronimo command	171, 173
local	41	RO—see REDON command	

RUNJOB (RJ) command	22, 42.3, 132	Teletypewriters	
RV—see RETRIEVE command		7023	40, 42.3
		7071	40
Saving background jobs	58	TEMPORARY property qualifier	29
Scheduler		Terminal devices	40
high level	57, 63	TERMINATE (TE) command	28, 42.5, 139
low level	57	Termination of listings	28
Scheduling	2, 57	Termiprinters	
Second error	62	7083/1	40
Secure magnetic tapes	32	7084/1	40
SECURITY command	2	7085/1	40
Sentinel	44	7086/1	40
SETUP file	28	Test programs, engineer	30
Setting up GEORGE		text parameter format	66
with GEORGE Executive	13	TH—see TAPERIGHT command	
with operator's Executive	9	TIME installation parameter	105
Siting of object programs in core	22.4	TP—see TAPES command	
SIZEDEFAULT installation parameter	105	TRACE Geronimo command	171
SK—see SPEAK command		TRANSPORTS (TR) command	34, 140
SLOTTIME installation parameter	105	TTTP—see tape-to-tape processor	
SPEAK command	42.2, 42.5, 133	TW—see TAPEWRONG command	
Special stationery			
output to	28	UNLOAD command	141
use with property system	31	Unjammer, backing store	46
Standard magnetic tapes	31	Unloading	
Start routines	43	basic input peripherals	25
STATE Geronimo command	171	basic output peripherals	28
Steering lines to		direct access devices	36
#XKOF	9, 21	magnetic tapes	35
STOP button, U.D.A.S. devices	36	Unusable peripherals	36
STOPLIST command	42.4, 42.5, 134	URGENCY command	58.1
Swap residences	7	Urgency level	58.1
System		username parameter format	65
breakdowns	59, 62.1		
closedowns	63	VIDEO command macro	143, 170
errors	59, 62.1	NOCHECK option	143, 173
requests	4, 157	virgin magnetic tapes	31
:SYSTEM.HLSEMERGENCY	57	Visual display unit, alphanumeric	169
:SYSTEM.HLSFILE	57	VOLUME installation parameter	105
:SYSTEM.IDF	22.2, 42		
:SYSTEM.INCINDEX	53	WA—see WHATPROPS command	
updating	53	WAITTIME installation parameter	105
:SYSTEM.JOBSTARTER	57	WAITWEIGHT installation parameter	105
:SYSTEM.OUTPUT	134	WHATJOBS (WJ) command	22.3, 58.1, 145
:SYSTEM.PROPERTY	22.1	WHATLIST (WL) command	42.5, 147
:SYSTEM.SERIAL	32	WHATPER (WP) command	24, 148
		WHATPROPS (WA) command	153
Tape		WHATSTATE (WS) command	22.3, 58.1, 154
fails and format errors	44	WJ—see WHATJOBS	
ICLKG3MARK _n	7	WL—see WHATLIST	
ICLKG4MARK _n	7	WLIST Geronimo command	171
magnetic	30 to 36	Work tapes	32
tape serial number parameter format	65	Workload	
Tape-to-tape processor (TTTP)	51	processing	22.3
end conditions	53	Worktape store	32
input tapes	52	WP—see WHATPER command	
messages	54	WRONG (WR) command	42.5, 156
operation	52	WS—see WHATSTATE command	
output tapes	52		
TAPERIGHT (TH) command	33, 136	Xenotape	31
TAPES (TP) command	32, 137		
TAPEWRONG (TW) command	33, 138		

#XK65	
error messages	62
program	61
#XK7S Journal Accounting Program	58.3
#XKOF	
error messages	9, 21
program	9, 21
steering lines	9, 21
#XPJ programs	9, 21

Amendment record

Should any additions or amendments to this Manual be necessary, they will be issued in the form of Amendment Lists.

When such changes have been incorporated, an appropriate entry should be made in the table below and the pink instruction sheet(s) should be filed behind this page.

[illegible]

