HALT (HT)

Function

Puts the object program into a halted state.

Format

```
HALT    text
```

The text may be up to 40 characters long. Excess characters will be lost. See *Texts,* page 163, for format details.

Forbidden contexts

NO CORE IMAGE, PROGRAM

(Note that HALT is not allowed when a break-in occurs during a command issued by an object program.)

Execution

The HALT command generates a pseudo program event of the category HALTED. The category of this event and the text, if any, specified in the command are written to the monitoring file system as output of the OBJECT category. In addition the category of the event and the text, together with the number of the current program member, are set up as the current program event message, overwriting the information stored by a previous program event (see *Program events,* page 22.2).

If HALT is issued during a break-in and the break-in occurred while the object program was running, this run of the program will be terminated. This means that when the break-in finishes, the job will return to command level as if a program event had occurred.

Example

HALT	A null text will be sent to the monitoring file system and set up as the current program event message.

Error messages

None

BUFFER OUT OF RESERVATIONS
CHARACTER ADDRESS NOT VALID
CONT FAILURE
CONTROL AREA OUT OF RESERVATIONS
COUNT OUT OF RANGE
END OF FILE VIOLATION
INVALID CONTROL AREA WORD *n*
MODE IN ERROR
MODE NOT AVAILABLE ON THIS UNIT
MODE NOT RECOGNISED
MODE SEQUENCE ERROR
MODES INCOMPATIBLE
MT HAS NO WPR
NO CSN IN WORD *n*
PERIPHERAL ALREADY OPENED
RESERVATION VIOLATION
WRITING NOT PERMITTED

(i) PARITY FAILURE

A parity failure within the object program area has occurred (only detected on certain processors).

9 MEMBER *number*

This condition is satisfied if the most recent program event occurred in the program member with the specified member number.

After any program event, the program can be continued by the use of a RESUME or ENTER command.

Miscellaneous conditions

The following are simple conditions which do not correspond to program events.

10 DISPLAY

The condition is satisfied if there is a display message set as the current display of the job.

11 DISPLAY *enclosed string*

As many characters as are given in the string are compared with the current display of the job.

12 REPLY

The condition is satisfied if there is a reply set as the current reply of the job.

13 REPLY *enclosed string*

As many characters as are given in the string are compared with the current reply of the job. The current reply is the last message output to the monitoring file system, excluding messages in COMMANDS and ONLINE categories; however, the exact wording cannot be guaranteed, a consideration which should be borne in mind when using the command. The delimiters of the 'enclosed string' determine whether spaces are to be significant in the comparison in the same way as for the other IF conditions. The number of possible replies is very large and reference should be made to the commands responsible for their issue for an appropriate list.

14 ON *number list*

This condition is satisfied if the on/off bits specified in the number list are on (see page 164 for the format of a number list). It is not satisfied if there is no core image or if a bit other than 0 to 23 is specified.

15 OFF *number list*

This condition is satisfied if the on/off bits specified in the number list are off. It is not satisfied if there is no core image or if a bit other than 0 to 23 is specified.

16 ABSENT *enclosed string*

This condition is satisfied if there is no parameter beginning with the specified character string in the command at the current level.

17 PRESENT *enclosed string*

This condition is the opposite of ABSENT. It is satisfied only if there is a parameter beginning with the specified character string.

18 STRING *enclosed string = enclosed string*

This condition is satisfied if the two character strings are identical. '=' may be replaced by '<' or '>', in which case the condition is satisfied if the strings match up to the end of the first (<) or second (>) string.

Note: The delimiters of the character strings must be consistent. Either they must both imply that spaces are to be significant in the comparison or they must both imply the opposite.

19 ZERO *number list*

This condition is satisfied if each of the numbers in the list is zero.

20 POSITIVE *number list*

This condition is satisfied if each of the numbers in the list is greater than zero.

21 NEGATIVE *number list*

This condition is satisfied if each of the numbers in the list is less than zero.

22 TRUE

This condition is always satisfied.

23 FALSE

This condition is never satisfied.

24 CORE

This condition is satisfied if one of the current contexts of the job is CORE IMAGE (see *Command contexts,* page 3).

25 MOP

This condition is satisfied if one of the current contexts of the job is MOP (see *Command contexts,* page 3).

26 USER

This condition is satisfied if one of the current contexts of the job is USER (see *Command contexts,* page 3).

27 FINISH

This condition is satisfied if a FINISH command has been issued previously and has not been cancelled.

28 EXISTS (*file description*)

This condition is satisfied if a file or secure magnetic tape can be found which corresponds to the file description given. No trap checking or checking on whether the file is user-frozen is performed.

29 RESTARTED

This condition is satisfied if the job has been restarted after having been running, or after having been in the well during a system reload with :SYSTEM.JOBLIST on an unavailable residence.

30 RESTARTED *enclosed string*

This is satisfied if condition 29, RESTARTED is satisfied, and if the qualification string, which must be one of the following is also satisfied.

(a) ABANDON

This is satisfied if the job has at some time been TEMPORARILY ABANDONED whilst running

(b) BREAK

This is satisfied if the job has at some time been running when a system break or closedown occurred, or was in the well when a system reload was performed with :SYSTEM.JOBLIST on an unavailable residence.

The qualification string may be abbreviated to any shorter length.

31 GENRES

This is satisfied if the job has been restarted from a well record which was dumped in an increment now being used to perform a general restore. This implies the job was in the system at the time the proper user's directory was dumped.

32 GENRES *enclosed string*

The string must be one of

(a) QUERY

This is satisfied if it is unknown what happened to the job following the dump from which the general restore is being performed. It will remain satisfied unless and until the operator issues a SCHEDULE command to set up one of the following conditions for the job.

(b) BREAK

This is satisfied if the operator issues a SCHEDULE command to inform the job that it was running at the break which led to the general restore.

(c) RERUN

This is satisfied if the operator issues a SCHEDULE command to inform the job it was completed after the dump but before the break which led to the general restore.

(d) RUNDUMP

This is satisfied if the operator issues a SCHEDULE command to inform the job it was running during the dump.

QUERY and BREAK may be abbreviated to any shorter length, but to avoid ambiguity, RERUN and RUNDUMP must always retain the first two characters.

Forbidden contexts

None

Execution

The condition is first evaluated and then, only if it is satisfied, *command*$_1$ is obeyed. For an IF command in Format 1, if the condition is not satisfied *command*$_2$ is obeyed. For an IF command in Formats 2 or 3, if the condition is not satisfied the command is not obeyed.

Examples

```
IF    HALTED(ERROR),GOTO    7
IF    DISPLAY'DI' AND OFF(3,7,19), GO TO    12
IF    STRING(ABC)<(%A),IF    NOT    MEM2,GO TO    21
IF    STR(%A)=(),(EN    1)ELSE(EN    8)
IF    NOT    DEL(OK),(GO TO    1ERROR)
IF    EXISTS(:FRED.TOM(1/B1)),(AS    *CR0,:FRED.TOM(1/B1))ELSE (OL    *CR0)
IF    MONITOR(OPEN *MT),GO TO    1LAB%;MESSAGE(15,16);
IF    NOT    MOP,EXIT
```

Error messages

ILLFORMED CONDITION (where the condition does not have the permitted format).

PARAMETER FORMAT ERROR (where the second parameter is not in a permitted format).

Notes:

1 No error is flagged if either

(a) the condition is satisfied and *command*$_2$ is wrongly specified, or

(b) the condition is not satisfied and *command*$_1$ is wrongly specified.

2 In Formats 1 and 2, if the *condition* is satisfied erroneous second parameters which start with (*command*) where *command* is a valid command will not be flagged as errors.

3 In some cases where it might be thought that an error message should be given there is in fact no syntax error. For example, the ON and OFF conditions may be used when no core image exists, in which case neither condition is satisfied.

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, except in the following cases:

1 If *parameter name* is JOBTRACE, MOPTRACE, OPTRACE or MINTRACE the *new value* parameter must be in the standard form for the *action on monitoring* file parameter list.

2 If *parameter name* is CONTEXT, the *new value* parameter must be 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.*

At Early Morning Start, all parameters are checked against their maximum and minimum values. For any that are outside this limit the initial value is taken. Some parameters, marked with * in the table, are also subjected to certain consistency checks. These checks are described in the manual *GEORGE 3 and 4 Operation Management.* Again, for any parameters failing these checks the initial value is taken.

Parameter name	*Initial value*	*Max.*	*Min.*	*Units*	*Meaning*
BACKGROUND	1	1	0	Jobs	Maximum number of background jobs that may be fully started.
BACKJAM	90	99	50	%	The proportion of a backing store residence that must be occupied before the backing store unjammer will take action.
BACKJOBS*	1	1	0	Jobs	Maximum number of background jobs that may be started.
BACKTHRESH	10	50	0	%	The proportion below BACKJAM at which the threshold of all non-special residences is set.
BSINTERVAL	300	$2^{23}-1$	0	Mins.	The target interval between entries to the backing store unjammer.
CLUSTERBACK	UNSET	0	0	Jobs	The target number of background jobs from clusters.
CLUSTERMOP	UNSET	0	0	Jobs	The target number of MOP jobs from clusters.

Parameter name	*Initial value*	*Max.*	*Min.*	*Units*	*Meaning*
CHAPTERQUOTA	15360	15360	0	Words	The amount of core store intended to be used by GEORGE chapters.
CONTEXT	0	A,B,C	0	Context bits	The context privileges that are currently in force.
COREOBJECT*	6144	6144	0	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	$2^{23}-1$	1	Mins.	The intervals at which data is collected by the system performance package (see the manual *Operation Management*) 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	2	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.

Parameter name	*Initial value*	*Max.*	*Min.*	*Units*	*Meaning*
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.
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*	2	2	0	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	LOGGING	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*	6144	6144	0	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 BROADCAS	ALL	LOGGING	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 jobs.
MOPJOBS*	1	1	0	Jobs	Maximum number of MOP jobs that may be started.
MOPLIMIT*	1	1	0	Jobs	Maximum number of MOP jobs that may be fully started.
MOPTOTAL	1	$2^{23}-1$	0	Jobs	Maximum number of MOP jobs that may be in the system, including the well.

Parameter name	*Initial value*	*Max.*	*Min.*	*Units*	*Meaning*
MOPTRACE	LOGGING and BROADCAS	ALL	NONE	Monitoring file categories	The initial setting of a MOP job TRACEing to the monitoring file.
MOPWAITTIME	5	$2^{23}-1$	1	Mins.	The delay after a MOP job time out before it is sent into limbo. For MOP jobs from VDUs, which do not time out, one minute is allowed after the last invitation to type before MOPWAITTIME begins to run.
OBJECTQUOTA*	0	0	0	Words	The amount of core store intended to be used by object programs. In GEORGE 4 this includes core for page turning and page tables.
OPTRACE	NONE	ALL	NONE	Monitoring file categories	Categories traced to central operator's console, in addition to messages that are sent in any case.
PERFWAIT	720	$2^{23}-1$	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	100	1	Mins.	The interval between each output of program and peripheral progress report messages to System Journal.
SIZEDEFAULT*	6144	6144	0	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).

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

JOBDATA RESOURCE(S) INCORRECT

The message ERROR IN JOB with no further comment may also occur when the internally issued INPUT command is in error. This message will be preceded by an INPUT error message.

Logging messages

STARTED *user name job name date time* TYPE *job type*

This message is sent to the monitoring file as the job leaves the well. A further part of the message is normally sent only to the System Journal. This is:

CONSOLE PROPERTY : *property name*, PARAM ($a_1,a_2,\ldots a_n$), *residual parameters*

CONSOLE PROPERTY : *property name* is only present for a cluster job, PARAM ($a_1,a_2,\ldots a_n$) is a copy of the PARAM parameters if present, and the residual parameters are any that would otherwise be unaccessed.

JOB RESTARTED AFTER BEING TEMPORARILY ABANDONED and/or LOST IN BREAK

For any job being restarted, this message is sent to the monitoring file and System Journal, preceding the STARTED message. Either or both of the qualifying clauses can appear, and indicate what happened to the preceding run.

JOB PRESENT IN SYSTEM AT DUMP FROM WHICH GENERAL RESTORE PERFORMED: *parameters*

For any job being restarted after a general restore, this is sent to the monitoring file and System Journal, preceding the STARTED message. The parameters are essentially determined by information supplied to the job by the operators on restart, and are a selection of

RUNNING AT BREAK

COMPLETED AFTER DUMP

RUNNING DURING DUMP

UNKNOWN ACTION AFTER DUMP

JOBDATA (JD)

Function

Stores or updates the scheduling requirements of a job in a file, which is consulted by the high level scheduler. This command is only for use in installations with a high level scheduler.

Format

1 JOBDATA *decimal integer, scheduling parameter*

The decimal integer indicates to GEORGE how the scheduling parameter(s) is to be handled. The table below gives the possible values of the decimal integer and the scheduling parameter(s) that must be specified in each instance.

Decimal integer	*Format of scheduling parameter(s)*	*Standard significance*	*Examples*
1	*mill time* If no alphabetic characters are given 'SECS' is assumed	Maximum job time	JOBDATA 1,4 JOBDATA 1, SECS4 JOBDATA 1,2MINS
3	*date*	Earliest start date	JOBDATA 3,1/1/73 JOBDATA 3,1JAN73
4	*time*	Earliest start time in hours and minutes or in hours, minutes and seconds	JOBDATA 4,10.40 JOBDATA 4,12.11.20
5	A string of up to four alphabetic characters, preceded or followed by a decimal integer less than 64.	On-line peripheral needs	JOBDATA 5,4CR JOBDATA 5, MT11
6	Decimal number	Maximum core size allowed for job	JOBDATA 6,33000

2 JOBDATA *scheduling information, scheduling information, . . .*

The scheduling information consists of an identifier followed by scheduling data which must be enclosed in parentheses if it includes any commas. The table below gives the possible identifiers and the scheduling data that must be specified in each instance.

Identifier	*Format of scheduling data*	*Standard significance*	*Examples*
UR	Letter	Urgency	JOBDATA UR M
JT	*Mill time* If no alphabetic characters are given 'SECS' is assumed	Maximum job time	JOBDATA JT 5MINS
DEADLINE	*Date, time, estimated run time.* The first parameter may be omitted	Deadline for job. (not used by standard HLS)	JOBDATA DEADLINE (1/1/75,09.30,10MINS)

Identifier	*Format of scheduling data*	*Standard significance*	*Examples*
STARTTIME	*Date, time*	Earliest start time	JOBDATA STARTTIME (1/3/77,10.50)
NEEDS	Strings (separated by commas) of up to four alphabetic characters, preceded or followed by a decimal integer less than 64	On-line peripheral needs	JOBDATA NEEDS 2MT JOBDATA NEEDS (1MT, 1 CR)
MZ	Decimal number K K is optional	Maximum core size allowed for a job	JOBDATA MZ 3000 JOBDATA MZ 4K
MQ (G4 only)	Decimal number K K is optional	Maximum quota allowed for a job	JOBDATA MQ 7K JOBDATA MQ 4096

The NEEDS identifier and the associated parentheses can be omitted when on-line peripheral requirements are specified.

The NEEDS identifier and the on-line peripheral requirements may be preceded by the operators + or –, in which case the data is added to or subtracted from the current scheduling requirements of the job. If a + or – operator occurs before a NEEDS identifier or an on-line peripheral needs items in a JOBDATA command, then all such items are treated as signed, with + assumed by default. If a NEEDS identifier is preceded by –, then all items to which it refers have their signs inverted. For example, –NEEDS (–MT1, +CR1, LP1) has the effect of adding one MT, subtracting one CR, and subtracting one LP from the already established requirements. However, if no + or – operator occurs, then the requirements expressed by the JOBDATA command completely replace any existing requirements. For example, JOBDATA 3MT, LP1 establishes the requirements as three MT and one LP, whatever has occurred previously.

MZ and MQ may also be preceded by the operators + or –, in which case the existing maximum size or maximum quota is augmented or decremented by the specified amount.

Forbidden contexts

NO USER

Execution

FORMAT 1

The decimal integer is used to index a table held in the first record of the :SYSTEM.JOBLIST file. This table provides a key indicating which built-in routines are to be used in processing the scheduling parameter(s). The table also gives the position in the relevant job entry in :SYSTEM.JOBLIST in which the scheduling data is to be stored. The data is stored in the following way:

Decimal integer	
1	The maximum job time is stored in a word of the job entry as a number of seconds
3	The earliest start date is converted into the number of days since 1/1/1900 and is stored in a word of the job entry
4	The earliest start time is converted into seconds and is stored in a word of the job entry
5	The alphabetic characters are converted by the internal routines to a peripheral type number and the integer is stored in binary form in the corresponding part of the NEEDS entry
6	The maximum size of core image allowed to the job is stored in the job entry as a number of words

FORMAT 2

The identifier is used to access a table within GEORGE. This table defines the format of the scheduling data and where, in the relevant job entry in :SYSTEM.JOBLIST, it is to be stored. The data is stored in the following way:

Identifier

UR — The urgency is stored in a word of the job entry.

JT — The maximum job time is stored in a word of the job entry as a number of seconds.

DEADLINE — The deadline date is converted into the number of days since 1/1/1900, the deadline time is converted into seconds and the run time is converted into seconds and they are stored in three consecutive words of the job entry.

STARTTIME — The earliest start date is converted into the number of days since 1/1/1900, the earliest start time is converted into seconds and they are stored in two consecutive words of the job entry.

NEEDS — The alphabetic characters are converted by internal routines to a peripheral type number and the integer is stored in binary form in the corresponding part of the NEEDS entry.

MZ — The maximum size of core image allowed to the job is stored in the job entry as a number of words.

MQ — The maximum quota of core image allowed to the job is stored in the job entry as a number of words.

The information stored by the JOBDATA command is used by the high level scheduler subject program in deciding which jobs to activate.

Notes:

1 Since high level scheduling is controlled by a subject program, the program may be changed to suit the special requirements of a particular installation. Other forms of the JOBDATA command may well be used by installation written subject programs. Full descriptions of how to do this may be found in Chapter 8 of the manual *GEORGE 3 and 4 Operation Management.*

2 The facilities of the JOBDATA command are used by a number of system macro commands. The macro commands that correspond to the different facilities of the JOBDATA command are as follows:

Macro command	*JOBDATA decimal integer*
STARTTIME	3 and 4
NEEDS	5

3 JOBTIME is equivalent to JOBDATA with a first parameter of 1.

Error messages

THERE IS NO SUCH JOBDATA COMMAND

z NOT RECOGNISED

PARAMETER FORMAT ERROR

PARAMETER NULL

z CONTAINS AN UNPAIRED DELIMITER

PARAMETER MISSING

JOBTIME (JT)

Function

Sets the job timer to the specified value.

Format

$$\text{JOBTIME}\quad \text{integer}\left\{\begin{array}{l}\text{SECS}\\ \text{MINS}\end{array}\right.$$

If neither MINS nor SECS is given, SECS is assumed.

Forbidden contexts

NO USER

Execution

This command overrides the default setting of the job timer set when the job was started (see the section *The job timer*, page 19).

Notes:

1 This command provides a safeguard against the possibility of job description errors causing wastage of a user's time budget, for example owing to an endless loop.

2 The command can be obeyed only once in a job.

Example

JOBTIME 20	A mill time limit of 20 seconds will be set on the job.

Error messages

COMMAND ISSUED PREVIOUSLY

PARAMETER FORMAT ERROR: *z*

TIME PARAMETER MISSING