Chapter 9 Handling Budgets

FUNCTIONS OF THE BUDGETING SYSTEM

The budgeting system enables the installation manager to control the use of his installation's resources and provides him with accounting information on users' activities. There are four types of activity which can be controlled by the budgeting system, and the four corresponding budget types are as follows:

1 MONEY A money budget is the quantity of money, real or notional, allocated to a user to pay for his use of computer time and space over a period of time fixed by the installation manager.

2 TIME (*urgency*) A time budget is the amount of mill time allocated to the user for running jobs and is subdivided into amounts of time for use on jobs of different urgencies (A to Z).

3 SPACEMT A space budget is the number of magnetic tapes available to the user.

4 REALTIME A realtime budget is the amount of core store available to the user for running realtime programs.

At the end of a budgeting period, the installation manager runs the accounting subject program to compile period accounts for each user and to allocate fresh transient budgets.

The ways in which the budgeting system affects the user are described below.

Initiating jobs

When a user starts a job with a DISCONNECT, JOB, LOGIN or RUNJOB command, his money budget is checked to ensure that he has not already used more than his allowance of MONEY. If he has, he will not be permitted to start the job and the error message

z HAS OVERDRAWN HIS MONEY BUDGET

will be given.

Provided that he is not actually overdrawn, he will be allowed to run the job, even though he may become overdrawn in the course of the job. Thus, a user with zero units available will be accepted. The reason for this is that some installation managers may not wish to use the money budgeting facilities. Since the amount of money available to users will then remain zero throughout each budget period, users will not be prevented from running jobs as a result of the budgetary controls.

Note: a DISCONNECT command causes the user's money budget to be checked only if he uses the command to start a new job.

When a user wishes to have his job made "fully started" the High Level Scheduler checks his time budget to see that he has time at the required urgency. If he has no time at this urgency, the High Level Scheduler will start the job at the next lower urgency at which he does have time. (This applies to the standard HLS, but may not apply at an installation that has written its own high level scheduler.)

Acquiring magnetic tapes

When a user issues a GET, GETONLINE or NEW command or an unanticipated open mode #400 PERI in order to bring a magnetic tape under his permanent ownership, his ration of tapes is checked to ensure that he is not already using his entire allocation. If he is, the command is rejected and the error message:

USER HAS NONE OF THIS BUDGET AVAILABLE

is given. If the user attempts to bring a number of tapes under his ownership by means of a single NEW command and his ration of tapes is sufficient for him to obtain some but not all of them, GEORGE will grant him as many of the tapes as his budget permits and will then output the above message. In this case the user can discover how many of the tapes he has succeeded in obtaining by issuing a LISTDIR command.

A user's budget of magnetic tapes does not include worktapes acquired by means of an ONLINE command or an open mode #600 PERI.

Running Programs with Realtime Requirements

If a user wishes to run a program with realtime requirements, he must have sufficient REALTIME budget for the size of the program concerned. When a REALTIME ON command is issued to switch a program to the realtime state, GEORGE checks that the user has enough REALTIME budget. If not, the error message

NOT ENOUGH REALTIME BUDGET

is output and the program is not switched to the realtime state.

SETTING BUDGETS

Rations

When a user creates new users by means of the MAKEDIR command, these new users initially have no budgets allocated to them and consequently they are not able to run jobs or acquire magnetic tapes. To make them operational the creator-user must allocate some or all of his budgets to them.

The standard method of allocating budgets is by means of the BUDGET command. This enables a user to give away rations of magnetic tape, time, money and realtime core space, to his immediate inferiors or, if he wishes, to his immediate superior. The format of the command is

BUDGET *user name, budget type, number*

In this case being considered, the user name will be the name of one of the users who have just been created. The budget type may be any one of the following:

MONEY	at the installation manager's discretion.
TIME (*urgency*)	measured in units of one second of mill time.
SPACEMT	measured in units of one magnetic tape.
REALTIME	measured in units of one word of core.

The number will be a positive number of units of the budget type specified and should not be greater than the ration at the creator-user's disposal.

Thus, the user might set up a new user and allocate budget rations to him with the following series of commands:

```
MAKEDIR     BRANCHA, :SMITH
BUDGET      :SMITH, SPACEMT, 20
BUDGET      :SMITH, TIME(M), 100
BUDGET      :SMITH, MONEY, 1000
```

The budget amounts manipulated by the BUDGET command are rations, not allowances. This means that though :SMITH will be entitled to 100 units of time as soon as the appropriate command has been given, he will have to wait until the start of the next budget period before he can actually start a job, since he has no money until then. In other words, transient budget allowances for the current period cannot be changed by means of the BUDGET command. Effectively this means that :SMITH in the example will not become operational until the start of the next budget period.

Allowances

If the creator-user wishes to make his new user operational at once, or if at any time he wishes to supplement another user's allowances, he must issue an ALLOWANCE command. This, as the name implies, acts only on transient budgets, that is, time and money. The format of the command is

ALLOWANCE *username, transient budget type, number*

Using this command, the creator-user can make :SMITH operational at once by allocating him money out of his own money allowance:

```
ALLOWANCE   :SMITH, MONEY, 200
```

As with the BUDGET command, the allowance transferred must not be greater than the unconsumed allowance at the creator-user's disposal.

Unlike the BUDGET command, the ALLOWANCE command enables a user to give away budgets to any user known to the system, not just to his immediate superior and his immediate inferiors.

ALTERING BUDGETS

Rations

In addition to increasing the rations of his immediate superior or of an immediate inferior by the method described above, a user can also reduce the rations of his immediate inferiors. This too is done by means of the BUDGET command. The format of the command is the same, except that the third parameter, which specifies the number of units to be transferred, must be a negative number. Thus, if the creator of BRANCHA, in the example, wishes to take back some of the SPACEMT and MONEY that he has allocated to :SMITH, he can do so by means of the following commands:

```
BUDGET      :SMITH, SPACEMT, -10

BUDGET      :SMITH, MONEY, -500
```

The alteration of :SMITH's money ration does not affect the money allowance available to :SMITH during the current budget period. The alteration of his ration of SPACEMT takes immediate effect and reduces the number of tapes he can bring under his ownership.

Allowances

In the same way the ALLOWANCE command can also be used to take back budgets, in this case transient budget allowances. Though a user can give away allowances to any user, he can take them back only from his immediate inferiors.

If the creator of BRANCHA finds that another of his inferiors, :JONES, is desperately in need of additional money for the current budget period, he may be able to help him by making an immediate grant from his own money allowance, by means of a positive ALLOWANCE command:

```
ALLOWANCE   :JONES, MONEY, 200
```

Alternatively, if his own money allowance is exhausted or not available, he can take back a part of the money allowance of an immediate inferior and give that to :JONES:

```
ALLOWANCE   :SMITH, MONEY, -200

ALLOWANCE   :JONES, MONEY, 200
```

A user can find out how much of any budget type and which privileges (see below) he or any of his immediate inferiors possesses, by issuing a BUDGETQUERY command.

PRIVILEGES

Privileges are granted and rescinded by means of the BUDGET command. The format of the command is

```
BUDGET   user name, privilege, GIVE or TAKE
```

If TAKE is specified, the privilege is rescinded; if GIVE is specified, it is granted. When a privilege is removed from a user, all that user's inferiors also lose the privilege.

There are several privileges built into the GEORGE budgeting system: NEWUSER, TRUSTED, CONTEXTA, CONTEXTB and CONTEXTC.

The NEWUSER privilege enables a user to create real users' (as opposed to pseudo users') directories, by means of the MAKEDIR command. When a senior user creates a new user, he must decide whether he wishes this user to have the power to create further users. Thus, the creator of :SMITH may decide to grant him this privilege:

```
BUDGET   :SMITH, NEWUSER, GIVE
```

Once he has the privilege, :SMITH can himself create users. He can then share out a part of his rations or allowances to them and can pass on the NEWUSER privilege to some or all of them. This expansion of the filestore hierarchy can be carried to any depth, but the senior user in the system can, by means of the budgeting commands at his disposal, keep a firm control over growth and expenditure.

The TRUSTED privilege enables the user to run programs including extracodes which can only be issued by object programs with a trusted status.

The CONTEXTA, CONTEXTB and CONTEXTC privileges are used in conjunction with the GIN macro FORBID and the installation parameter CONTEXT to restrict the number of built-in commands available to certain users. The installation manager can use the FORBID GIN macro to forbid the use of any built-in command to users who do not have any or all of the privileges CONTEXTA, CONTEXTB and CONTEXTC. Including a FORBID GIN macro to set CONTEXTA, CONTEXTB or CONTEXTC context bits when loading GEORGE has in itself no effect on users, but if the installation manager then sets the installation parameter CONTEXT to A, B or C or any combination of the three, users not holding the appropriate CONTEXT privilege will be forbidden use of the command. If any such users attempt to issue the command, the following error message will be output:

ERROR IN *command*: YOU HAVE NOT THE PRIVILEGE TO USE THIS COMMAND

Note: Commands issued in NO USER context (except JOB, RUNJOB and LOGIN) will obviously not be restricted in this way.

It should be noted that privileges CONTEXTA, CONTEXTB and CONTEXTC are picked up from the Dictionary at the beginning of each job and remembered throughout the job. This is for efficiency reasons. So, for example, if :MANAGER were to NAMEPRIV CONTEXTA halfway through a job he would still not be able to issue commands forbidden in that context during that job's run.

Another privilege, NOTOWNER, is built in to GEORGE and is initially held only by :MANAGER. It cannot be passed on to users by the BUDGET command until the manager has entered it in the Dictionary using the NAMEPRIV command. Further information on the use of the privilege is given in the manual *GEORGE 3 and 4 Operation Management.*

Installation managers use the NAMEPRIV command to define privileges to suit the particular needs of their installations (see also Chapter 10 of the manual *GEORGE 3 and 4 Operation Management*).

ACCOUNTING

GEORGE provides accounting systems at two levels as follows:

1 LOG ANALYSIS The log analysis systems run regularly (depending on which system is used) and provide charging information on completed jobs. The installation manager is offered a choice of log analysis systems. In particular the Journal Accounting Program processes information stored in the System Journal and charges each job for its use of core, processor and peripherals. The manager can also write his own program to process accounting information held in the System Journal and could write programs to construct accounts based on the sizes of users' files and the numbers of magnetic tapes they own. If an installation does not require comprehensive and flexible accounting, an alternative log analysis system is provided, the built-in system. This system is quick and simple and just charges for the mill time used by each job. (A third log analysis system, the built-out system, is still provided but is likely to be withdrawn in the near future. In its unmodified form it is compatible with the Journal Accounting Program so nothing further will be said about it.)

2 PERIOD ACCOUNTING The period accounting program is run at the end of each budget period (say, each month). It compiles period accounts for each user and allocates fresh budget allowances for the next period.

Built-in log analysis

Built-in log analysis processes the job's monitoring file in accordance with the parameters of the terminating command and calculates the charge for the amount of mill time used.

There are three budget lines at the end of the listing. They are ordinary LOGGING category messages and printing of them can be suppressed. Their format is as follows:

BUDGET	USED	LEFT
TIME (*urgency*)	–	–
MONEY	–	–

There is only one TIME line output, giving time used and time left at the last urgency specified in the job.

Journal Accounting Program

The Journal Accounting Program calculates the charge for running a job on the basis of System Journal messages of the LOGGING category. All LOGGING category messages are sent to the System Journal and at fixed intervals

the Journal Accounting Program processes messages in the Journal to produce accounts for all jobs that have finished since the Program last ran.

The Journal Accounting Program converts resources used on a job into money by means of an algorithm (described in the manual *GEORGE 3 and 4 Operation Management*). The total money charged and the total mill time used at each urgency are subtracted from the user's money and time budgets.

The fixed interval at which the Journal Accounting Program runs is 2 hours in the standard Program but individual installations may vary the interval to suit their own needs. It can be seen therefore that users will not be able to obtain up-to-date accounting information immediately at the end of each job.

A list of the logging category messages that are of interest to the Journal Accounting program is given below.

LOGGING MESSAGES

The following logging messages may be sent to the monitoring file and, in packed format, to the System Journal, in the course of the job.

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

2 GEORGE 3 only: *time milltime* CORE GIVEN *number,* CLOCKED *progtime* (see Note 1 below)

3 GEORGE 4 only: *time milltime* SIZE GIVEN *number,* CLOCKED *progtime* (see Note 1 below)

4 GEORGE 4 only: *time milltime* SIZE GIVEN *number* SPARSE, CLOCKED *progtime* (see Note 1 below)

5 *time* USED *number* AS *peripheral name*

6 *time* FREE *peripheral name, number* TRANSFERS

7 *time* FREE *number, number* TRANSFERS

8 GEORGE 3 only: *time milltime* DELETED, CLOCKED *progtime*

9 GEORGE 4 only: *time milltime* DELETED, CLOCKED *progtime* MAXQ = *number*K, PT = *number,* USED *number*K (see Note 3)

10 *time mill time* USED URGENCY *letter,* CLOCKED *progtime* (see Note 1 below)

11 *time mill time* REALTIME STARTED, CLOCKED *progtime* (see Note 1 below)

12 *time mill time* REALTIME FINISHED, CLOCKED *progtime* (see Note 1 below)

13 *time mill time* FINISHED: $number_1$ LISTFILES HERE, $number_2$ ELSEWHERE AND FURTHER LISTFILES FOR $number_3$ MULTIFILES (see Note 2 below)

These messages are generated by the following commands and extracodes:

1 DISCONNECT, LOGIN, JOB, RUNJOB

2,3,4 CORE, LOAD, LOADENTER, RESTORE, RESUME, SIZE, GIVE/4 and GIVE/12 extracodes

5 ONLINE

6,7 ASSIGN, DELETE, LOAD, LOADENTER, ONLINE, RELEASE, RESTORE, RESUME, REL extracode, Close mode PERI

8,9,12 DELETE, ENDJOB, LOAD, LOADENTER, LOGOUT, QUIT, RESTORE, RESUME, DEL, DELTY

10 LOAD, LOADENTER, ENTER, CREATE, ASSIGN, ONLINE, RESUME, SAVE, MONRESUME

11,12 REALTIME

13 ABANDON, CONNECT, ENDJOB, LOGOUT

Notes:

1 the 'CLOCKED *progtime*' part of the message is output only if *progtime* is non-zero

2 relevant parts of the message will be omitted if $number_2$ or $number_3$ is zero

3 See Appendix 1, *GEORGE 4.*