The Resource Manager

University of Michigan Information Technology Division Consulting and Support Services

Contents

Intr	oduction	3
1.1	What is the Resource Manager?	3
1.2	How Does the Resource Manager Work?	3
1.3	What is a Job?	4
1.4	What is a Job Queue?	4
1.5	What is a Pseudodevice?	5
1.6	Additional Documentation and Assistance	5
Bate	h Processing	6
2.1	What is a Batch Job?	6
	2.1.1 Reasons for Using Batch	6
2.2	Submitting a Batch Job	
	2.2.1 Entering Commands in a File	7
	2.2.2 How Long Does Batch Take?	8
	2.2.3 Monitoring a Batch Job	8
	2.2.4 Logging Batch Output	9
	2.2.5 Viewing Batch Output	9
	2.2.6 Cancelling a Batch Job	9
	2.2.7 Common Errors with Batch	10
2.3	Changing Batch Defaults	11
	2.3.1 Job Selection and Rates	11
	2.3.2 Estimating Time and Pages	13
2.4		
	1	
	2.4.2 Changing Print Options	
	1.1 1.2 1.3 1.4 1.5 1.6 Bato 2.1 2.2	 1.2 How Does the Resource Manager Work? 1.3 What is a Job? 1.4 What is a Job Queue? 1.5 What is a Pseudodevice? 1.6 Additional Documentation and Assistance Batch Processing 2.1 What is a Batch Job? 2.1.1 Reasons for Using Batch 2.2 Submitting a Batch Job 2.2.1 Entering Commands in a File 2.2.2 How Long Does Batch Take? 2.2.3 Monitoring a Batch Job 2.2.4 Logging Batch Output 2.2.6 Cancelling a Batch Job 2.2.7 Common Errors with Batch 2.3 Changing Batch Defaults 2.3.1 Job Selection and Rates 2.3.2 Estimating Time and Pages 2.3.3 Assigning a Job Name 2.3.4 Mounting Magnetic Tapes 2.3.5 Setting Print Routes and Delivery Codes 2.4 Printed Output For Batch Jobs 2.4.1 Head Sheet, Tail Sheet, and Job Statistics

		2.4.3 Cancelling Hardcopy Output	
	2.5	Other Things You Should Know	
		2.5.1 Reruning Batch Jobs	19
		2.5.2 Conditional Sign-off	
		2.5.3 Sigfile Processing	20
		5 5	
3.	Gett	ing Printed Output	
	3.1	What is *PRINT*?	
	3.2	When to Use \$COPY and \$LIST	
	3.3	Printing Options	
	0.0	3.3.1 Setting Print Options for the Entire Session	
		3.3.2 Setting Print Options for the Current Print Job	
		3.3.3 Setting Print Options Using \$9700	
		3.3.4 Summary of \$SIGNON, \$SET, \$CONTROL, and \$9700	
	3.4	Print Routes and Delivery Codes	
	$3.4 \\ 3.5$	Multiple Copies	
	3.6	Cancelling a Print Job	
	3.0 3.7	Creating Your Own Print Device	
	3.1 3.8	Local Printers	
	3.0	Local Printers	
4	Ъ	naging Batch and Print Jobs	20
4.		The \$CANCEL Command	
	4.1		
	4.2	The \$CONTROL Command	
	4.3	The \$CREATE Command	
	4.4	The \$DISPLAY Command	
		4.4.1 Displaying \$SET Command Keywords	
		4.4.2 Displaying Pseudodevice Names	
	4.5	The \$LOG Command	
	4.6	The \$RELEASE Command	
	4.7	The \$SET Command	
	4.8	The \$SYSTEMSTATUS Command	
		4.8.1 The USERS Keyword	
		4.8.2 The QUEUE Keyword	
		4.8.3 Explanation of Responses	
		4.8.4 When is a Job Purged?	
	4.9	The \$VIEW Command	
5.	The	*RMLOG Program	52
	5.1	PRINT Command Options	
	5.2	Time/Date Ranges	
	5.3	Record Types	
	5.4	Examples	56
Ap		ix A: GDINFO Codes and Sense Data	
	GDINFO Type Codes		
	Sens	se Data	58
Ind	lex		59

1. INTRODUCTION

This document is intended as an introductory guide to using the Resource Manager. We assume that you are familiar with the material presented in the *Introduction to MTS—Getting Started*, T7003. In particular, you should know how to use basic MTS commands, edit files, and run simple programs.

Each MTS mode uses a prompt character that is displayed in the first column of your terminal or printer output. These prompt characters indicate which part of MTS you are talking to:

- # MTS operating system
- : MTS File Editor (\$EDIT)
- ? Several programs use this to prompt for text

This prompt or prefix character appears in the first column of many of the examples; you *should not* type it as part of your input.

The following conventions are used for the command notation throughout this document:

UPPERCASE	indicates material to be repeated verbatim by the user.
lowercase	represents a value which is to be replaced by an item supplied by the user.
brackets []	represents a parameter or value that is optional and may be omitted.
braces { }	indicates that the material within the braces represents choices, of which one must be selected. The choices are separated by vertical bars.

1.1 WHAT IS THE RESOURCE MANAGER?

The Resource Manager (RM) is a system that was developed to handle all the facilities associated with printing and batch processing on a large mainframe computer. It is an integral part of the Michigan Terminal System (MTS) and is invoked automatically whenever you initiate a job that performs any of these types of processing.

1.2 HOW DOES THE RESOURCE MANAGER WORK?

Whenever you initiate an operation that requires one of these facilities:

- (1) page, local, or line printers
- (2) batch facility
- (3) BITNET file transfers
- (4) BITNET messages

the Resource Manager does the following:

- (1) assigns a job number and job name in order to keep track of your job;
- (2) assesses the requirements of your job, based on certain defaults, the size of the job, and information supplied on \$CONTROL and \$SIGNON commands;
- (3) assigns a priority based on this assessment, and passes your job to the appropriate queue to await processing;
- (4) releases your job for processing when the required device or facility is available;
- (5) directs output to the device indicated by your job (if this is a printer, it recognizes keywords you might have used to modify the manner of printing);
- (6) removes a job from the system upon completion or cancellation;
- (7) retains a record of your job for a period of time;
- (8) maintains a log of your job from start to finish.

In the sections that follow, you will see what commands are available to initiate and modify jobs of various types.

1.3 WHAT IS A JOB?

Throughout this document we refer to a *job* as the focal point of the Resource Manager. What do we mean by a job? As used in this context, a job is a distinct action initiated by you, the user, that results in a separate and complete process. For example, if you want to print a file called MYFILE you could initiate a job by typing:

\$LIST MYFILE ON *PRINT*

After you enter your \$LIST command, the Resource Manager sends you a message indicating that it has received and processed your job:

PRINT assigned job number 135568 *PRINT* RM135568 released to CNTR 10 images 5 sheets RATE=NORMAL

The Resource Manager also assigns your job a unique name and number. In this case your job name is RM135568 and your job number is 135568. At this point your job, which totals 10 pages to be printed on 5 sheets of paper, has been sent to the print queue (more about queues later) and is waiting to print. The four characters CNTR indicate that your file will be printed at the Computing Center station on North Campus.

1.4 WHAT IS A JOB QUEUE?

When the system is not very busy, you might find that your jobs are processed almost as quickly as you can submit them. Most of the time, however, there are a lot of other users competing with you for the system's resources. The Resource Manager assigns a priority to each job depending on the type and amount of resources required, then places your job in a *job queue* where it must wait until those resources are available and all those of higher priority have been processed.

There are several queues available for different types of jobs. A print job that is generated in a conversational terminal session will go directly to a *print queue*. A batch job must first go to the *batch queue*. Then, when it finishes the execution step, any printed output it generates must wait in a print queue.

1.5 WHAT IS A PSEUDODEVICE?

We have described the Resource Manager as a system that manages jobs that require the use of certain facilities. Each of these facilities is described by a pseudodevice name; specifically, we have:

BATCH	for submitting a batch job.
PRINT	for producing printed output.
PUNCH	for producing 80-byte record output.
IMPORT	for importing data using BITNET connections.
EXPORT	for exporting data using BITNET connections.

Pseudodevice names are a convenient way of referring to a class or type of device. For example, we have many different printers available for use: the Xerox 9700 Page Printer and the Hewlett-Packard LaserJet, to name only two. Rather than assigning a name to every printer that we have, we assign a general name (called a pseudodevice name) that we can use to refer to any print device. Then when you need to print a file, you can decide which printer you will actually use (the details of how this is done will be described later). It is important that you understand the concept of pseudodevices as they will be used throughout this document.

1.6 ADDITIONAL DOCUMENTATION AND ASSISTANCE

If you need assistance in printing with the Resource Manager, there are many sources of help.

The following documents give additional information about the Resource Manager: Using the Xerox 9700 Page Printer, Reference R1038. BITNET on MTS, Reference R1039.

For additional assistance on MTS, see the *Introduction to MTS*, Tutorial T7003, and *MTS Volume 1: The Michigan Terminal System*, Reference R1001.

Consulting assistance is available over the phone by calling 764–HELP. If you want information on walk-in consulting, refer to the file *CONSULTINGHOURS to find out when and where you can see a consultant. You can also use the MTS Message System to SEND TO ONLINE_CONSULTING to ask questions about the RM.

2. BATCH PROCESSING

This section describes how to use the MTS batch job facility. It covers general concepts, and presents some basic examples for running a batch job using *BATCH*. This is followed by an explanation of the available sign-on parameters, printed output, and by other useful information.

2.1 WHAT IS A BATCH JOB?

A batch job is simply a set of commands that are processed by the computer without interactive control. In other words, once you have submitted the job you cannot change the sequence of flow, nor correct any errors during execution. You sacrifice the flexibility available in *interactive mode*, and you may have to wait several hours, or even overnight, to receive your results. Before deciding to use batch you should ask yourself the following questions.

- (1) Do I know the exact steps required to complete the job?
- (2) Can I run the program without being prompted for input from the terminal keyboard?
- (3) Am I confident that the job will run without errors? Often batch jobs will exceed time or page limits (see "Common Errors with Batch" in this section).
- (4) Am I prepared to wait (possibly overnight) to get the results back from my job?

If the answer is "yes" to the above questions, then you are ready to use batch.

2.1.1 Reasons for Using Batch

There are two basic advantages for you in using batch mode: the first is cost, the second is time.

When you run your job interactively, you are in effect requesting the use of computer resources: for example, printers or disk space. Interactive mode means that the computer attempts to respond to your request as quickly as possible while at the same time responding to similar requests from all other users on the system. Batch mode allows the computer to take its time in responding and allocate resources to your job when they are not required by interactive jobs. What this means for you is perhaps a longer wait for completion of your job, but at a reduced cost. This reduced cost is due to the fact that you are not charged for terminal connect time and that your job may run at a time when the rates are lower. A more detailed explanation of the charges for batch processing is given in "Job Selection" in this section.

The second advantage in using batch is that under certain circumstances batch allows you more efficient use of your time. This is best illustrated by an example. Let us assume you have a job that typically consumes a lot of elapsed time (for example, running *RESTORE to reinstate several files). Rather than sit idly at a terminal waiting for execution to terminate, you might submit a batch job and get on with other work while it is being processed. If you require a *hardcopy* record of your run, this is supplied automatically in batch, and includes such pertinent information as time and accounting charges.

In addition to the above situations, any job that is processed at regular intervals in a standard manner is a good candidate for batch processing. These types of jobs are frequently referred to as "production jobs".

Remember that while you are waiting for your batch job to complete, you are free to continue working on other tasks, such as editing or submitting jobs, provided they do not use files required by your batch job.

2.2 SUBMITTING A BATCH JOB

The easiest way to submit a job to batch is to copy the commands directly to *BATCH*.

Let us assume you were producing a course catalog, using a program in the file CATMAKER to generate your output, and you want to run your job in batch because it will cost less. The course catalog is in an MTS file called CATALOG. You can submit your batch job directly from *SOURCE* (normally the terminal keyboard) to the pseudodevice *BATCH* as follows (remember, you should not type the prompt characters):

```
# $COPY *SOURCE* *BATCH*
> $SIGNON * PAGES=100
 *BATCH* assigned job number 273907
> $RUN CATMAKER INPUT=CATALOG PRINT=*PRINT*
> $SIGNOFF
> $ENDFILE
# *BATCH* RM273907 released HOST=UM RATE=NORMAL
```

When you enter the line containing your \$SIGNON command, you should always use an asterisk (*) instead of your userID if you are running the batch job from the same userID. This avoids the security risk of typing your password at your terminal. The PAGES=100 parameter requests a limit of 100 pages for your job (the default limit is 50). You may then type in any commands you wish to execute in batch; e.g., running CATMAKER.

After entering the first line, the system assigns your batch job a name and number. Your job name is RM273907 and your job number is 273907. Note that your job name is simply the job number with the letters RM prefixed to it. You can assign your own job names (more about this in "Assigning a Job Name" in this section). You can use your job name to keep track of the progress of your batch job (more about this in "Monitoring a Batch Job" in this section).

You terminate the batch job by typing the line \$ENDFILE or by entering an end-of-file at the ">" \$COPY prompt. Your job will then be released.

All batch jobs produce printed output which is sent directly to *PRINT*. Since you did not change the default for *PRINT*, the output from the job will be printed on one of the Xerox 9700 page printers. Details on printed output will be covered in "Printed Output for Batch Jobs" in this section.

2.2.1 Entering Commands in a File

If you expect to run the job several times, you can put the commands in an MTS file and copy the file to *BATCH*.

Using your previous example, create a file called CAT.RUN and use the editor to add commands to the file.

\$EDIT CAT.RUN
: INSERT
? \$SIGNON * PAGES=100
? \$RUN CATMAKER INPUT=CATALOG PRINT=*PRINT*
? \$SIGNOFF
: STOP

Now, any time you want to submit this as a batch job, just type:

\$COPY CAT.RUN *BATCH*

Your job will be submitted to the batch queue. In general, we recommend that you use this second technique for submitting batch jobs for two reasons: first, it is easier to correct errors; second, if you are submitting the same job several times, it will be more efficient than typing in the entire command sequence each time.

2.2.2 How Long Does Batch Take?

Often a batch job will begin executing immediately after it is submitted; however, this is not always the case. The time required for a job to complete is the *turn-around time*. Turn-around is affected by a number of factors including the number of users on the system, current jobs executing or in the queue, availability of resources such as printers, and the priority level of the job submitted. Because of this the turn-around for a particular job can vary significantly. The following are a few guidelines to make note of:

- (1) A job that is submitted with the parameter RATE=MINIMUM on your \$SIGNON command will usually run overnight (more details in "Job Selection" in this section).
- (2) Jobs requiring large amounts of computer time or resources will, in general, have a slower turn-around time.
- (3) Before submitting your job, it is possible to check current activity on the system using the \$SYSTEMSTATUS command (see the section "Managing Batch and Print Jobs").
- (4) The system will *not* notify you when your job has finished; however, you can check using the \$SYSTEMSTATUS command. This will be explained in the next section.

2.2.3 Monitoring a Batch Job

Once you have submitted your batch job, you may check its progress.

Assume you have submitted two jobs to *BATCH* and want to see if they are finished. You do not know what their names or numbers are so you issue the command

\$SYSTEMSTATUS QUEUE *

or more simply

SYQ*

The system will respond with the following:

1ABC:RM316636 is awaiting execution, P7, after 26 jobs. 1ABC:RM316638 was executed at 10:48:52 Sun Jan 14/90 1ABC:RM316639 is awaiting print, P13, after 21 jobs.

The first line refers to a batch job still awaiting execution; there are 26 jobs preceding it in the queue. It has an execution priority of 7 (indicated by P7), which is a medium priority. The second entry tells us that job RM316638 has been executed. The last entry refers to the printed output created by the second job which is waiting to be printed, there are 21 jobs ahead of it in the print queue. Note that \$SYSTEMSTATUS QUEUE * will display all of your jobs currently waiting in a queue plus any jobs that have completed execution in the last eight hours.

Another form of the \$SYSTEMSTATUS command which can be used as follows:

```
$SYSTEMSTATUS QUEUE USER (or $SY Q U)
```

This form of the command will only display those jobs that are currently waiting to be executed. You can use this form if you are certain that your job has not been executed.

2.2.4 Logging Batch Output

Once your job has been executed, the output will be sent to *PRINT*. Further details on printed output are given in the section "Printed Output for Batch Jobs" in this section.

Normally output from a batch job is printed. However, if you also want the output of your batch job to be put into a permanent file, then you should include the following commands at the start of your batch job:

```
$SIGNON *
$EMPTY MYLOG
$LOG *MSINK* ON MYLOG
.
.
.
.
(enter batch commands here)
.
$SIGNOFF
```

Note that you will need to have created the file MYLOG before you submit the above job; the results of your batch run will then be placed into the file MYLOG.

2.2.5 Viewing Batch Output

You may use the \$VIEW command to look at the command lines of a batch job before it is executed or at the results of the job before they are printed (see the section "The \$VIEW Command").

2.2.6 Cancelling a Batch Job

You can cancel a job submitted to *BATCH* using \$CANCEL provided, of course, that the job has not been already executed. For example, suppose you have submitted a job with a low priority and subsequently decide that you want to submit it interactively rather than wait for the results. First, obtain the job name:

\$SYSTEMSTATUS QUEUE *

The system responds,

1ABC:RM316636 is awaiting execution, P8, after 26 jobs.

Then you enter,

\$CANCEL RM316636

You can then check again,

\$SYSTEMSTATUS QUEUE *
1ABC:RM316636 was cancelled at 15:53:02 Sun Jan 14/90.

Further details on using \$CANCEL are given in the section "Managing Batch and Print Jobs".

2.2.7 Common Errors with Batch

A batch job will terminate abnormally if it requires more central processing unit (CPU) time or pages than you have allowed. In this case, you will find the message:

GLOBAL item LIMIT EXCEEDED

on the last page of your output, where *item* could be one of either TIME or PAGES. When you submit a job to batch there are a number of optional parameters that you can specify using \$SIGNON. If you do not specify these, the system will automatically assign default values for you: 3 seconds for execution (CPU) time and 50 pages for printed output. If you expect to exceed these limits, be sure to specify appropriate values with the \$SIGNON command. It is wise to estimate higher than you expect.

For example, if you are submitting a job that requires about 8 seconds of CPU time and normally prints 376 pages, you could define reasonable limits such as:

\$SIGNON * PAGES=400 TIME=10

These are slightly higher than your requirements to allow for a small margin of error, but not so high that the priority of the job would be adversely affected.

There are two reasons why it is not a good idea to request an excess of either time or pages to ensure that your job will execute. First, time and page restrictions are an insurance to you that your job is not out of control producing unwanted pages of output while using up precious computer dollars; second, excessive time limits will lower your job's position in the execution queue and thus increase the turn-around time and may even cause the job to be held by the operator and not run. For more details on estimating TIME and PAGE requirements, see "Estimating Time and Pages" in this section.

Regardless of the limits you define on your \$SIGNON command, you will only be charged for the amount of resources actually used. However, if you request a limit that would exceed the funds remaining in your account—even if your job does not actually require that much—the system will *not let your job sign on*. The job will abort.

Typing and syntax errors, as mentioned earlier, are also quite common. These are more easily corrected when batch jobs are submitted by entering your commands into a file. Note also that although the use of the dollar sign is optional (i.e., \$SIGNON and SIGNON are equivalent), we recommend that you always use them. If an error is detected during execution, MTS will look for the next line in your file beginning with a "\$" and continue from there.

2.3 CHANGING BATCH DEFAULTS

Unless you explicitly modify the attributes of a batch job, it will execute subject to the following criteria:

- (1) The job is executed at normal priority.
- (2) It is printed in landscape format on the Xerox 9700 page printer, on standard 8 1/2 by 11 paper.
- (3) There is a 3-second global limit for execution (CPU) time and a 50-page global limit for printed output.
- (4) All commands and system responses are echoed in the the printout.

You can change these default options by using the \$SIGNON command. The first line of your batch job contains a \$SIGNON command with either a userID or an asterisk (*). The general form of this command is:

\$SIGNON * parameters

or

\$SIGNON userid parameters

The first form, which uses the asterisk, can only be used when you are submitting a job from your own userID. If you are submitting a job for a userID other than the one you are currently signed on to, then you must use the second form. Note that if you are using the second form, you will also need to provide the password for the userID that you are using. Generally, most users will find the first form easier to use. In both cases parameters are optional and can be one or more selected from the list in Table 2. The following is a simple example:

\$SIGNON * RATE=MINIMUM PAGES=200

In the above example you will be signed on to run a job with minimum rates (RATE=MINIMUM) with a limit of 200 pages of printed output (PAGES=200). The order in which \$SIGNON parameters are specified is not important. Note that there should be no blank spaces before or after the "=" sign.

The \$SIGNON command is described more fully in *MTS Volume 1: The Michigan Terminal System*, Reference R1001. The remainder of this section discusses some the \$SIGNON parameters that are of particular interest to batch users.

2.3.1 Job Selection and Rates

Both the job priority and the rate period are used to select jobs for execution and print processing.

The priority of the job is the position of the job within the execution or print queue.

The job execution priority is assigned as an integer between 1 and 10 and is based on the CPU time estimate specified on the \$SIGNON command. The job print priority is an integer between 0 and 15 and is based on the *actual* number of pages (images) to be printed. The following table summarizes the priority assignments for the NORMAL rate period:

Priority	Execution	Images
Class	Time (seconds)	Printed
15	•••	≤ 5
14	•••	≤10
13	•••	≤20
12		≤50
11	•••	≤100
10	≤1	≤200
9	≤ 2	≤ 350
8	≤ 4	≤ 500
7	≤8	≤650
6	≤16	≤800
5	≤ 32	≤1000
4	≤ 64	≤ 1500
3	≤ 128	≤2000
2	≤ 256	≤4000
1	>256	≤6000
0		>6000

Table 1: Priority Assignments for Jobs

Jobs are selected in decreasing order of priority class and, within any given class, in a first-in, first-out (FIFO) basis. The page estimate does not affect the priority calculations for that job.

The rate period specifies the rate at which the job will be charged. Jobs for which RATE=LOW, RATE=DEFERRED, or RATE=MINIMUM is specified are automatically held for execution during the low-, deferred-, or minimum-rate periods. Such jobs are charged at a reduced rate. Jobs for which RATE=NORMAL is specified are eligible for execution at any time.

The rate period schedule is as follows:

Low Rates (45%)	6 pm - 10 pm Monday through Thursday
Deferred Rates (65%)	10 pm - 12 am Monday through Thursday 12 am - 2 am Tuesday through Friday 6 pm - 10 pm Friday 7 am - 6 pm Saturday and Sunday
Minimum Rates (80%)	2 am - 7 am Tuesday through Friday 12 am - 7 am Friday to Saturday 6 pm - 7 am Saturday to Sunday 6 pm - 7 am Sunday to Monday
Normal Rates	All other times

The percentages given above reflect the CPU-rate discount that is applied to sessions run during these periods. Other resources may have differing discounts. Student course accounts on the UB system are assigned a single rate that is in effect all the time.

Terminal jobs are always executed at the lowest available rate at sign-on time. The rates for terminal jobs are determined by the sign-on time and remain in effect for the duration of the session.

The rate for a terminal job may not be changed by specifying the RATE option on the \$SIGNON command.

Batch jobs are executed at the current rate at sign-on time if the RATE specification is omitted from the \$SIGNON command. A higher or lower rate period may be specified by using the RATE option. If a higher rate period is specified, the job will be executed before other batch jobs of lower rate periods. If a lower rate period is specified, the job will be held until the specified rate period is in effect. However, if the job cannot be executed during the next available lower rate period, it will be executed as soon as possible thereafter (i.e., it will not be held over until the lower rate period recurs).

Batch jobs and *PRINT* jobs submitted at LOW, DEFERRED, or MINIMUM rates may be printed anytime, but are assigned priorities lower than would be the case for similar-sized, NORMAL-rate jobs. The priority decrements are as follows:

LOW-rate jobs	2 lower than NORMAL
DEFERRED-rate jobs	3 lower than NORMAL
MINIMUM-rate jobs	4 lower than NORMAL

For example, a 50-page job would be priority 12 for NORMAL rates, 10 for LOW rates, 9 for DEFERRED rates, and 8 for MINIMUM rates.

The public file *RATES gives the current rate structure for each of the different classes of accounts. You can obtain this information by issuing the command

\$COPY *RATES

2.3.2 Estimating Time and Pages

The amount of time and pages required is tied closely to the type of job being run; however, there are a few general guidelines to follow when making your estimates. If you have run the job previously using either batch or interactive mode, the last page of your listing will usually have an accounting of the pages printed and time used—you should refer to this for approximations.

If you are currently running a job interactively, you can obtain an estimate of the CPU time required by noting the CPU time used which is given on the "Execution terminated" message.

Note the following when you are estimating pages: the Resource Manager counts a single sheet, printed on both sides, as two pages of output. If you are using a format that prints two images on a page, then a single sheet would contain 4 pages of output. Also, remember to include all of the hardcopy that you are directing to *PRINT* in your estimates.

If you are submitting a large job which you anticipate could run for several minutes (or even hours), then do some smaller test runs first to get an accurate estimate. Finally, remember that if a job aborts due to exceeding time or page limits, *you will be charged for resources used*. Therefore, if you do not know what to estimate, get some advice from the ITD Consultants (764–HELP) instead of submitting a job several times on a trial and error basis using up computer dollars in the process.

2.3.3 Assigning a Job Name

You can uniquely identify your batch job by assigning a job name by issuing a command of the form

\$SIGNON * **JOBNAME**=jobname

The job name is a string of up to eight alphanumeric characters, starting with a letter; for example, ANDY or EPA001. This is useful if you have jobs that you submit on a regular basis. It also makes it easier to remember your job name when you are using the \$SYSTEMSTATUS command.

You must use care when assigning names to jobs that are submitted on the UB-MTS system. All printed output from those jobs is assigned a new job number when it is transferred to the UM-MTS system for printing. Thus if you have several jobs with the same name, it may be difficult (or impossible) to distinguish those jobs since you will not know the new job number assigned after the transfer.

2.3.4 Mounting Magnetic Tapes

If your batch job requires magnetic tapes to be mounted, use the TAPES option on the \$SIGNON to improve the chances that sufficient tape drives will be available when your job starts:

\$SIGNON * TAPES=n

However, even with this, your job may start executing when all the drives are in use. In this event, the operator must stop your job and rerun it later, which could have undesired results (see "Reruning Batch Jobs" in this section). Also, you should use a single \$MOUNT command to mount all tapes that must be simultaneously available.

2.3.5 Setting Print Routes and Delivery Codes

The ROUTE option determines where your output is printed; the DELIVERY option implies that your job will be delivered to another site on campus. Either of these options can be altered with your \$SIGNON command. A more complete explanation is given in the section "Getting Printed Output".

Table 2: Options Available on \$SIGNON

Option

Description

ADDRESS="line1;line2;..."

Specify the campus mail address for delivered output (when DELIVERY=MAIL is set). This option applies only to page-printer output, not to line-printer or local-printer output.

COPIES=n Specify number of copies of printed output. The default is n=1.

DELIVERY={station | MAIL | NONE}

Specify a delivery station to which output should be delivered by messenger service (see the file *DELIVERY for further details). If MAIL is specified, the output will be delivered by campus mail or the US Postal Service. The default is NONE.

FORMAT={PORTRAIT | LANDSCAPE | TWOUP | format-name}

Specify format for page-printer output (see *Using the Xerox 9700 Page Printer*, Reference R1038, for a complete description and list of *formats*). The default is LANDSCAPE.

JOBNAME={jobname | DEFAULT}

Assign a job name of 1 to 8 alphanumeric characters to the current job. The first character must be a letter. DEFAULT specifies the default format of "RM" plus six digits.

MARGIN={n.nn | NO}

Set the left margin to "n.nn" inches. "n.nn" must be less than the current page width (8.5 for portrait orientation, 11.0 for landscape). MARGIN=NO resets the margins to the default for the current format (0.5 for PORTRAIT and 0.65 for LANDSCAPE).

{ONESIDED | TWOSIDED} Specify printing on one or both sides of the paper for page-printer output. The default is TWOSIDED.

OVERLAY={NONE | SHADED | LINED}

Specify an overlay for page-printer output (see *Using the Xerox 9700 Page Printer*, Reference R1038, for a complete description of overlay options). The default is NONE.

PAGES=n Specify maximum number of printed pages for job. The default is n=50.

PAPER={PLAIN | 3HOLE | LABEL24 | LABEL33}

Specify paper type for Xerox 9700 page-printer output. The default is PLAIN. The obsolete ANY option is synonymous with PLAIN.

{PORTRAIT | LANDSCAPE | TWOUP}

Specify orientation of page-printer output (synonymous with FORMAT for the corresponding values). The default is LANDSCAPE.

PRINTER={PAGE | LINE}

Specify the type of printer. The default is PAGE.

RATE={NORMAL | LOW | DEFERRED | MINIMUM} Specify job rate group for execution.

RERUN={YES | NO}

Specify whether job should be restarted if it is stopped during execution for any reason (see "Reruning Batch Jobs" below). The default is YES.

ROUTE=station Specify a print station for output (use the command \$COPY *ROUTE to obtain a list of valid station codes).

SEPCOPY={YES | NO}

Specify whether each copy will have separate head and tail sheets (YES) or whether all copies will be printed together as one job with a single head and tail sheet (NO). The option is only effective for page-printer output and if COPIES=n is specified. If each copy is more than 50 pages, then SEPCOPY=YES is forced. Currently the default is SEPCOPY=YES. On December 2, 1991 the default will be changed to NO.

- SHIFT={YES | NO}Specify whether page-printer output is shifted away from the binding edge. The default is SHIFT=NO.
- SIGFILE={ON | OFF}

Specify whether sigfile processing is to be in effect. The default is ON. More information on sigfiles is given below.

- TAPES=nSpecify number of tape drives required at any one time. Although this is
not absolutely necessary, it is highly recommended as it will improve the
chances that tape drives are available when your job starts executing.
- TIME=n Specify maximum CPU time in seconds. The default is 3.
- TWOSIDED={YES | NO} Specify printing on one or both sides of the paper for page-printer output. The default is YES.
- WAITUNTIL="time and/or date"

Specify when job will be run. Job will not begin sooner than specified time, but may run some time after depending on the rate group specified. For example, "January 10, 1990" or "15:30 January 10, 1990".

"comment" An arbitrary comment, enclosed within quotes, may be used to identify the job.

2.4 PRINTED OUTPUT FOR BATCH JOBS

Every batch job creates printed output. In some cases this output may only include a head sheet, a tail sheet, and job statistics. Typically, the output consists of the above plus a log of the batch session. This log contains all of the commands that were submitted to *BATCH* plus the system responses to these commands. By default, all printed output from the job is directed to *PRINT*.

2.4.1 Head Sheet, Tail Sheet, and Job Statistics

The first and last pages of the printed job are called the *head sheet* and *tail sheet*. These are used to identify each job, and to separate it from those that printed immediately before and after. The head sheet is printed in very large type and includes: your userID, the job name and number, the time and date it printed, and the delivery code. The tail sheet contains the same information in a much smaller format and also includes a summary of job statistics such as: CPU time, disk storage, lines read, pages printed, and so on for the current job. It also lists accounting charges for each resource used by the job, and gives an approximation of the your remaining balance.

Suppose you submitted a job by using *BATCH* as follows:

```
# $COPY *SOURCE* *BATCH*
> $SIGNON * PAGES=100
*BATCH* assigned job number 629384
```

> \$RUN CATMAKER INPUT=CATALOG PRINT=MYTEXT SERCOM=MYLIST
> \$SIGNOFF
> \$ENDFILE

BATCH RM629384 released HOST=UM RATE=NORMAL

This is what you might expect your job statistics to look like:

ID:		1ABC	
Project Number:		1AAA	
Signon Time:		11:57:06 Tue Jan 16/90	
Signoff Time:		11:57:33 Tue Jan 16/90	
Last Signon Time/Date	:	08:13:12 Mon Jan 15/90	
Charging Rate:	Batch	Normal, Univ/Gov't, UM, IBM 3090	-600E
		_	
Elapsed Time		minutes	
Cpu Time		seconds \$.50	
Active VM Integral		page-minutes	
Wait VM Integral		page-hours	
Lines Read	3	lines	
Page Printer Pages	3	pages	
Page Printer Lines	125	pages	
Page Printer Image	3	images \$.12	
Page Printer Sheet	2	sheets \$.02	
Disk I/O	394		
Computing Cost		\$.64	
Disk Storage	5760	page-hours \$.30	
Remaining Balance		\$927.04	
Job name:		RM629384	
Job number:		629384	
Host:		UM	
Devicetype:		9700	
Twosided:		YES	
Format:		LANDSCAPE	
Paper:		PLAIN	
Entered from AF1C at:		11:57:07 Tue Jan 16/90	
Printed on PTR2 at:		12:26:59 Tue Jan 16/90	

The above example tells you a number of things about your job. The job ran at 11:57 and was charged at normal rates. The total elapsed time was .067 minutes whereas it only used .726 seconds of CPU time. Elapsed time is always higher than CPU time due to the fact that even when a job is executing it will spend a certain amount of time waiting for resources, disk I/O, etc. The job read 3 lines; these are the 3 lines that were submitted to *BATCH* from your terminal. There were 3 images printed on 2 sheets of paper. All of the other output was put into permanent files, as indicated by the parameters PRINT=MYTEXT and SERCOM=MYLIST from the original job. If you had directed all of your output to *PRINT*, you would then expect the lines printed to be much higher. The job was finished at 11:57 and was printed at 12:26. The total cost for the job was \$.99. If you had printed the files MYTEXT and MYLIST, there would have been additional charges.

2.4.2 Changing Print Options

There are a number of techniques for changing your printed output. You can alter the print options for your entire batch job on the \$SIGNON command (see Table 2 for details). This is typically done for setting global options; for example, you can set the delivery station using the DELIVERY option.

The other techniques that you will probably find useful is creating multiple print streams within your batch job. This will allow you to print several files each with different print options within a single batch job. A full explanation of creating user defined pseudodevices (multiple print streams) is provided in the section "Getting Printed Output". The following is a simple example to illustrate how you might use this technique:

```
$SIGNON * PAGES=200
$CREATE *LIST* TYPE=PRINT OPEN='FORMAT=TWOUP'
$CREATE *TEXT* TYPE=PRINT OPEN='PORTRAIT ONESIDED'
$RUN CATMAKER INPUT=CATALOG PRINT=*TEXT* SERCOM=*LIST*
$SIGNOFF
```

This batch job consists of three separate print jobs. The first job is the log of your batch run which is directed to *PRINT*. The second print job is your program listing which will be printed in two-up format; that is, there will be two pages printed on each side of the sheet. Finally, the third print job is the output from your program run which will be printed on one side of the page (ONESIDED) in portrait orientation (PORTRAIT).

All of the options that are available for regular print jobs also apply to batch print jobs. Further details and examples of altering the appearance of your printed output are provided in the section "Getting Printed Output".

2.4.3 Cancelling Hardcopy Output

You can eliminate hardcopy output from your batch job by using the \$CANCEL command. For example, suppose you want to run the previous job with the listing and CATMAKER output placed into permanent files. The only printed output would be the job statistics which you could then cancel as follows:

```
# $COPY *SOURCE* *BATCH*
> $SIGNON * PAGES=100
 *BATCH* assigned job number 629384
> $RUN CATMAKER INPUT=CATALOG PRINT=MYTEXT SERCOM=MYLIST
> $CANCEL *PRINT*
> $SIGNOFF
> $ENDFILE
# *BATCH* RM629384 released HOST=UM RATE=NORMAL
```

You should use this form of \$CANCEL carefully, as it will eliminate *all* printed output from your job including error messages.

2.5 OTHER THINGS YOU SHOULD KNOW

Let us quickly look at a few more features which might be used to your advantage. These include:

- (1) reruning batch jobs
- (2) conditional sign-offs
- (3) sigfile processing

2.5.1 Reruning Batch Jobs

If there is a system failure (or a system shutdown where the operator decides to stop your job), then all jobs that are currently executing will be rerun *from the beginning*. This could present a serious problem if you had already emptied or destroyed input files that were used during an earlier part of your job. If you are adding data to the end of a file, the same information could be added twice, which may not be the desired effect.

Your job will also be automatically rerun if it requires tape drives that were not available while it was running. This problem is less likely to occur if you specify the TAPES option on the \$SIGNON command and mount all tapes using a single \$MOUNT command. Since tape mounts are not allowed on the UB-MTS system between 10 a.m. and 4 p.m., jobs submitted during that time requiring tape mounts should specify the WAITUNTIL option on the \$SIGNON command.

In order to protect yourself against possible problems in the event of a system failure, design your batch job so that it can be rerun from the beginning. If you create a file within the job, empty it right after creation so that it will not contain anything if the job is rerun.

When a job is rerun, whatever was done before the interruption is *not* undone when it restarts. You are not explicitly informed when a rerun has occurred, although you might find some clues in your batch job listing. For example, if you get the following system response while trying to create a new file:

```
File NEWFILE already exists
```

it is probable that your job was rerun.

One consolation is that you are only charged for the final complete run, and not for the aborted first attempt. If you do not want your job to be rerun, you have the option of specifying RERUN=NO with your \$SIGNON command.

2.5.2 Conditional Sign-off

In a batch job, all commands are processed sequentially regardless of previous errors. However, you can use the \$IF command after any \$RUN command to test for an error condition. Then, depending on the value of the *return code* set by the program, you can decide whether to sign off or continue. This does not offer as much flexibility as interactive computing, but it does let you terminate a job that is running into trouble.

The following is a simple example:

\$IF RUNRC > 4, \$SIGNOFF

The return code from the last executed program is tested. If it is greater than 4, the user will be signed off the system. Thus, if a program gives an indication of success or failure through values of the return code, this type of test may be used to abort a batch job if the program fails to execute properly.

You can also conditionally \$RUN another program based on the success/failure of previous steps. A complete conditional facility is available by using MTS command macros which are described in *MTS Volume 21: MTS Command Extensions and Macros*, Reference R1021.

2.5.3 Sigfile Processing

In MTS, it is possible to set up a special file called a *sigfile* that will contain a series of commands to be automatically processed whenever you sign on. You can check to see if you have a sigfile by issuing the command

\$DISPLAY SIGFILE

If MTS responds with the name of a file that is your current sigfile. You should print out a copy of this file and familiarize yourself with its contents. If you do not already have a sigfile, the rest of this section will explain briefly how you can set one up.

The following steps will create a sigfile called SIGS:

\$CREATE SIGS
\$EDIT SIGS
: INSERT
? \$SET ROUTE=NUBS
? \$SET COST=ON
?

This file will route all your printed output to NUBS and will also turn on the MTS cost display option. The next step is to identify this file as your sigfile by issuing the following command:

\$SET SIGFILE=SIGS

The next time you sign on to your userID (and every time thereafter), the two commands in SIGS will automatically be executed. If you decide that you want to *turn off* sigfile processing for all subsequent signons, issue the following command:

\$SET SIGFILE=OFF

This is a simple example that is intended to introduce the concept of sigfile processing. It is also possible to run a program from within your sigfile which will execute a series of commands *conditionally*, depending on such factors as what type of device you are signed onto, or what day of the week it is. If you want to know more, please see the description of *SIGSETUP in *MTS Volume 2: Public File Descriptions*, Reference R1002. A further explanation of sigfile processing is also given in *MTS Volume 1: The Michigan Terminal System*, Reference R1001.

3. GETTING PRINTED OUTPUT

Printed output can be produced during a terminal session or as a result of submitting a batch job. The various terms *printout*, *hardcopy*, or simply *output* are most often used to describe the printed results. Earlier we discussed some of the attributes that apply specifically to batch jobs; the following will cover the details of creating a print job from either a terminal or batch session.

3.1 WHAT IS *PRINT*?

In the introduction to this document we described pseudodevices and how they are used. *PRINT* refers to the pseudodevice that is used to produce your printed output. By default, *PRINT* is defined as one of the Xerox 9700 page printers.

You create printed output in a variety of ways. You can use the \$LIST command:

```
$LIST filename ON *PRINT* (the ON is optional)
```

or if the named file contains carriage-control characters (see *MTS Volume 1: The Michigan Terminal System*, Reference R1001), you can use \$COPY:

\$COPY filename TO *PRINT* (the TO is optional)

or you can send output directly from a job that is executing, e.g.,

\$RUN *PASCALVS INPUT=MYPROG OBJECT=MYOBJ PRINT=*PRINT*

In all of the above cases the Resource Manager follows the same sequence of events. It assigns the job a number and, if you do not assign one explicitly, a default job name. It evaluates the requirements of the job, and then submits it to the print queue for the designated device. Once in the queue the job will:

- (1) print immediately, or
- (2) print after waiting for jobs that were before it in the queue, or
- (3) print when the resources are available (e.g., label stock is loaded into the Xerox 9700 page printer).

If the print device is not functioning when the job is submitted, the Resource Manager will maintain a queue of all jobs until the machine is back in order. This can sometimes cause quite a backlog when the system is busy. The operations staff have the option of holding back very large print jobs, especially if the queue is long. A print job that generates a large amount of output may be printed in smaller segments. Of course, the user picks it up as one complete job.

3.2 WHEN TO USE \$COPY AND \$LIST

Some files contain special characters called carriage-control characters in the first column. These special characters act as instructions to the printer to control vertical spacing on the page. These are

usually inserted by text-processing programs; however, they can be added by the user. A common mistake when getting a printout of a file is to use the \$COPY command when the file does not contain carriage-control characters. The result is often that the output may be only one or two lines per page, thus generating an unexpectedly large volume of output. The clearest indication that you have this problem is when the first character of some (or all) of the lines is missing in the printed output.

The easiest way to check for carriage-control characters is to examine the file using the editor and check the first character of each line. If the first character is part of the actual text or data, *do not* use \$COPY to send that file to the printer; use \$LIST instead. On the other hand, if the first character of each line is a blank or a carriage-control character, the printer will use it to control spacing as indicated. Further information about carriage-control characters is given in *MTS Volume 1, The Michigan Terminal System*, Reference R1001.

The \$LIST command also is preferable if you are printing files with long lines since they will be wrapped onto the next line. With the \$COPY command, they will be truncated instead.

3.3 PRINTING OPTIONS

All output is sent to the printer with certain default printing options in place. If you do nothing more than direct output to the printer, your output will have the following attributes:

- (1) orientation is landscape
- (2) typeface is Xerox 1200 fixed-pitch
- (3) printing is on both sides
- (4) printing with no overlay
- (5) printing is not shifted from the "binding" edge
- (6) paper is 8 1/2 by 11, standard weight, plain (not 3-hole punched)

All of these options can be altered using the \$SIGNON, \$SET, \$CONTROL, or \$9700 commands. The choice of which command to use depends on the options required, and how long you want them to be in effect. It is not practical to illustrate here all of the ways that you can change your print options. However, we will discuss each of the four commands briefly and give an example of how they might be used.

3.3.1 Setting Print Options for the Entire Session

The \$SIGNON command can be used to set options for an entire session in batch mode. Setting global print options for interactive mode is done using \$SET. If you are interested in setting print options for batch processing, you should read the section on the \$SIGNON command entitled "Changing Batch Defaults".

The \$SET command is used to set print options for the remainder of the session—that is, once you have set your options they are in effect until you reset them or sign off. The main advantage of using \$SET is that you have the option of putting your commands into a file called a sigfile and then have that file automatically executed each time you sign on. The following is a simple example of how you might

use this technique.

Suppose you want all of your output to be printed with the page oriented in portrait (like this document) instead of in landscape (which is the default format). Assuming that you have a sigfile called MYSIGFILE, you would include the appropriate \$SET command as follows:

```
# $EDIT MYSIGFILE
```

```
: INSERT
```

? \$SET FORMAT=PORTRAIT

Now the next time you sign on (and for all sign-ons thereafter), all of your output will have portrait orientation. If you want to change this, you can reset the option using:

```
$SET FORMAT=LANDSCAPE
```

More details on creating and testing sigfiles is given the section "Sigfile Processing" earlier in this document. Sigfile processing is also useful for setting print routes and delivery codes (more about these in a later section).

3.3.2 Setting Print Options for the Current Print Job

You can use \$CONTROL to set print options for a single print job. For example, suppose you were printing a special report and you want to use paper that is lined to make it more readable. The output for your report is stored in a file called REPORT1.

After you enter the \$CONTROL command, the Resource Manager assigns a job number. You are also notified that your job is being *held*. What this means is that nothing will be printed until you issue \$RELEASE *PRINT* or sign off. The Resource Manager displays the specified options once the job has been released.

You can also use \$CONTROL to change several options at once. For example, if you want to print a file on 3-hole punched paper at Michigan Union Station (UNYN), you would enter the following:

The \$CONTROL command can also be used to change the format of your output. For example, to print a file using portrait orientation set with a shaded overlay:

\$CONTROL *PRINT* FORMAT=PORTRAIT OVERLAY=SHADED
 PRINT assigned job number 357596
 PRINT RM357596 held
\$COPY REPORT1 TO *PRINT*

\$RELEASE *PRINT*
PRINT RM357596 released to CNTR 10 images 5 sheets RATE=NORMAL
FORMAT=PORTRAIT OVERLAY=SHADED

A complete list of formats and overlays is available in *Using the Xerox 9700 Page Printer*, Reference R1038.

When you use \$CONTROL be aware that each time you issue the command the options you set will affect everything that you have printed but not yet released. For this reason we suggest that you always release your print job when you are finished with a particular print option.

If a line of the form

\$9700CONTROL option ...

appears in a file that is copied to *PRINT*, then it will act as if a \$CONTROL *PRINT* command with the specified options was issued. For example, if the file contains

\$9700CONTROL ONESIDED PAPER=PLAIN
\$9700 PORTRAIT MARGIN=0 FONTLIST=(AA1AJP,...)

then copying this to *PRINT* will act the same as if you first issued the command

\$CONTROL *PRINT* ONESIDED PAPER=PLAIN

and then copied the file containing the line

\$9700 PORTRAIT MARGIN=0 FONTLIST=(AA1AJP,...)

to *PRINT*.

Note that only legal \$CONTROL command options may be specified on the \$9700CONTROL line. As with the \$CONTROL command, all this does is set up the conditions that will be in effect at the *start* of the print job. If the same option is used more than once on \$9700CONTROL lines in a print job, the last one specified will be the one in effect.

If do not explicitly \$RELEASE your print job, it will be automatically released when you sign off. Also once you issue \$RELEASE, all of your print options are changed back to their default values with the exception, of course, of those that were changed using \$SET.

If you want to find out if a print job is currently being held, you can issue the command

\$DISPLAY *PRINT*

The message

PRINT RM357604 routed to CNTR 20 images 10 sheets RATE=NORMAL

will be printed indicating that you have 10 sheets of output waiting to be printed. If you decide that you do not want your printout, you can \$CANCEL it instead of issuing \$RELEASE (more about this in a later section). More examples of using \$CONTROL are given in the section "Managing Batch and Print Jobs".

3.3.3 Setting Print Options Using \$9700

The \$9700 command differs from previous commands since it is used to change print options *within the data being printed* as opposed to changing the print options for an entire job. In this way, you can alter the printed appearance of your job on a line-by-line or a page-by-page basis.

The \$9700 commands are inserted directly into the data stream instead of being issued from MTS. They will allow you to change options such as typeface, fonts, and page orientation. The use of \$9700 commands is explained in detail in *Using the Xerox 9700 Page Printer*, Reference R1038. If you are interested in altering print options in this manner, you are advised to read this document, or alternatively you should consider using one of the text-processors available on MTS that automatically insert the appropriate \$9700 commands.

3.3.4 Summary of \$SIGNON, \$SET, \$CONTROL, and \$9700

When you submit a batch job, you can assign your print options for the entire session using the \$SIGNON command. If you want a particular print option to always be in effect for either batch or interactive mode, you should create a sigfile and put in the appropriate \$SET commands. If you want to specifically change the print options for a single job, the \$CONTROL command is the one to use. The \$9700 command is quite specialized; its use should not be attempted without further reading as noted above.

A table of \$SIGNON parameters is given above in "Changing Batch Defaults". Similar tables for \$SET and \$CONTROL can be found in the section "Managing Batch and Print Jobs".

3.4 PRINT ROUTES AND DELIVERY CODES

You can use ROUTE option to specify where you want your output printed; the DELIVERY option implies that your job will be printed at one location and then delivered to another site by messenger. Let's look at two simple examples to illustrate the use of these options.

Suppose you are working in the Chemistry Building and you want to have your output printed at the nearby North University Building Station (NUBS). You enter your commands as follows:

```
# $SET ROUTE=NUBS
# $EMPTY MYOBJ
# $RUN *PASCALVS INPUT=MYPROG OBJECT=MYOBJ PRINT=*PRINT*
 *PRINT* assigned job number 629384
 *PRINT* RM629384 released to NUBS 18 images 9 sheets RATE=NORMAL
```

The listing produced by the compiler *PASCALVS will be printed at NUBS.

Alternatively, suppose you are working on the U-M Dearborn Campus and want the Dearborn messenger to pick up the output for you at the Computing Center. You would submit your job as follows:

Note that in the above examples that the ROUTE and DELIVERY options will be in effect for the entire session (this is because you used \$SET). If you want to change ROUTE or DELIVERY for a particular job, you would use \$CONTROL. If you are submitting a job in batch, you can change the print route or delivery code using \$SIGNON (see "Changing Batch Defaults").

A complete list of available print routes and delivery codes is maintained online; they can be obtained as follows:

\$COPY *ROUTE \$COPY *DELIVERY

If the DELIVERY=MAIL option is specified, one of two things will happen:

(1) For Xerox 9700 page-printer output, if the ADDRESS option is specified on either the \$CONTROL or \$SET command, the output will be delivered via campus mail. For example:

```
$SET DELIVERY=MAIL ADDRESS="Jon Dough;ITD;5074 Fleming 1340"
```

(2) For line-printer output, or for page-printer output that does not specify the ADDRESS option, the output will be sent to the ITD Mail Librarian for delivery via the US Postal Service at a cost of \$7.50 for each mailing in addition to actual shipping costs. The charge will be billed to your userID. To make arrangements for mail delivery, send your job number and mailing address to:

ITD Mail Librarian 535 W. William Street Ann Arbor, MI 48103

or send an electronic message via the MTS Message System to Mail_Librarian.

Output from the local printers is never delivered, even if the DELIVERY option is specified.

3.5 MULTIPLE COPIES

If you want to print more than one copy of your entire job (not including head and tail sheets), use the COPIES option with the \$CONTROL command and specify the number desired.

If you require each copy to have a head and tail sheet (i.e., to print as a separate job), include the SEPCOPY=YES option on the \$CONTROL command:

\$CONTROL *PRINT* COPIES=3 SEPCOPY=YES

Note that if each copy is larger that 50 pages, then each copy will always have a head and tail sheet, since in this case each copy is separately scheduled for printing.

The SEPCOPY option applies only to page-printer output. Line-printer output and local printer output are always printed with SEPCOPY=YES in effect.

3.6 CANCELLING A PRINT JOB

It is possible to cancel a job that you have submitted for printing if it is in the queue or is currently printing. First, get the job name and number as follows:

```
$SYSTEMSTATUS QUEUE *
```

The system responds,

1ABC:RM316640 (316640) is awaiting print, P8, after 10 jobs.

Then you enter,

\$CANCEL RM316640

The system will respond with

Job 1ABC:RM316640 cancelled.

As you can see your output was successfully cancelled. If you have not released *PRINT*, use \$CANCEL as follows:

\$CANCEL *PRINT*

When you use \$CANCEL as shown in the last example, you will be refunded the printing portion of the original charges for your output. Further details on using \$CANCEL are given in the section "Managing Batch and Print Jobs."

3.7 CREATING YOUR OWN PRINT DEVICE

Occasionally you may find that you are changing print options frequently. If this is the case, you might find it useful to create your own print pseudodevices (the concept of pseudodevices was explained in the "Introduction") with the options that you use for particular types of print jobs. It is possible to do this using the \$CREATE command.

For example, suppose you are working on a paper at the Church Street computing site (CHUR). The paper is in an MTS file called THESIS. You want to produce two types of printed output: the first is a file listing that you want printed on the Xerox 4045 local printer at the site; the second is your final text output, with portrait orientation, printed on the Xerox 9700 page printer at the Michigan Union Station (UNYN). You can do this by creating two new pseudodevices called *DRAFT* and *FINAL* as follows:

```
# $CREATE *DRAFT* TYPE=PRINT OPEN="ROUTE=CHUR"
 *DRAFT* has been created
# $CREATE *FINAL* TYPE=PRINT OPEN="FORMAT=PORTRAIT ROUTE=UNYN"
 *FINAL* has been created
# $LIST THESIS ON *DRAFT*
 *DRAFT* assigned job number 357400
 *DRAFT* RM357400 released to CHUR 12 pages
# $COPY THESIS TO *FINAL*
```

```
*FINAL* assigned job number 357401
*FINAL* RM357401 released to UNYN 11 images 6 sheets RATE=NORMAL
FORMAT=PORTRAIT
```

The output from your job has been sent to the specified printers. You are also free to use *DRAFT* or *FINAL* anytime again in the current session. If you want to use *DRAFT* and *FINAL* without explicitly creating them for each session, you should use \$CREATE in a sigfile to define them.

When you create print pseudodevices, you specify the attributes by using the OPEN option. Any options that are available with the \$CONTROL command may also be used with OPEN. It is also possible to specify several options at once provided they are separated by blanks:

\$CREATE *ONE* TYPE=PRINT OPEN="FORMAT=PORTRAIT ONESIDED"

You have created a print device called *ONE* which has the attributes of PORTRAIT and ONESIDED.

You cannot create a pseudodevice using a name already defined by the system or a name you created earlier in your session. To see all currently defined pseudodevice names (including predefined ones), use the command:

\$FILESTATUS *?*

If you want to remove the definition for a PDN that you have created (if you created it with the wrong values on OPEN, for instance), you can use the \$DESTROY command. Once you have destroyed it, you can recreate it, and specify different options.

3.8 LOCAL PRINTERS

Several Campus Computing Sites have Xerox 4045 or Hewlett-Packard LaserJet local printers. These are small page printers which are slower, have fewer built-in fonts, and may be restricted in other capabilities such as printing on only one side of the paper. They are best for printing short file listings, draft copies of documents, and copies of messages.

The Xerox 4045 and H-P LaserJet only support the PORTRAIT, LANDSCAPE, MARGIN, and OVERLAY=LINED \$CONTROL options; they do not recognize \$9700 command lines.

To send output to a local printer, you must specify the ROUTE parameter on the \$CONTROL or \$SET command

\$CONTROL *PRINT* ROUTE=station

For example, to print the file named DRAFT in portrait orientation on the Xerox 4045 at the Dana Building Computing Site:

\$CONTROL *PRINT* PORTRAIT ROUTE=DANA
\$COPY DRAFT *PRINT*

The following local printers are available at the Campus Computing Sites.

Route Code	Printer Type	Location
AROX	Xerox 4045	Angell Hall Courtyard
CHUR	Xerox 4045	611 Church, 4th Floor
DANA	Xerox 4045	Dana Building (Natural Resources)
FRZE	Xerox 4045	Frieze Building
SPH2	Xerox 4045	School of Public Health
UGLS	Xerox 4045	Undergraduate Library

Local printing to the Xerox 4045 and H-P LaserJet printers is charged for at the same rates as for the Xerox 9700 page printers.

4. MANAGING BATCH AND PRINT JOBS

The Resource Manager enables you to modify certain features of any job you submit for processing. It also provides a mechanism for keeping track of jobs once they have been submitted. This section contains a summary of the MTS commands that can be used.

\$CANCEL	removes a job from the processing queue.
\$CONTROL	changes the attributes of a specified pseudodevice.
\$CREATE	lets you define pseudodevice names.
\$DISPLAY	shows the current state of a pseudodevice or set item.
\$LOG	puts a record of your session in a file.
\$RELEASE	releases a job to the execution or print queue.
\$SET	changes certain global conditions that apply to a job.
\$SYSTEMSTATUS	keeps track of jobs that have been submitted during the last 8 hours.
\$VIEW	lets you view the results of a batch or print job.

The rest of this section is divided into parts, each describing one of the above commands. Many of these commands have already been introduced in this document; this section is intended as a reference for their use. More detailed information can also be obtained from *MTS Volume 1: The Michigan Terminal System*, Reference R1001.

Note: Most MTS commands may be abbreviated. However, for clarity abbreviations are used sparingly here.

4.1 THE \$CANCEL COMMAND

The \$CANCEL command lets you remove a job from the processing queue. You can cancel a job that is not done, i.e.,

- (1) executing or waiting to execute
- (2) printing or waiting to print
- (3) held by a \$CONTROL command

The general form of the command is:

\$CANCEL {jobname | jobnumber} [USER=userid]

or

\$CANCEL {*PRINT* | *BATCH*}

The first format is used for jobs that have been released for processing. In this case you must know the *job name* or *job number* (these terms were explained in the "Introduction"). Your job name and job number can be obtained by using the \$SYSTEMSTATUS command. If the job was submitted from a different userID, you must specify that userID with the USER parameter; you will be prompted for the password.

The second form is used when the job is still being held (i.e., you have not released it). In this case you must cancel the appropriate pseudodevice name (e.g., *PRINT*).

The effects of cancelling a job are as follows:

- (1) If the job has already been executed and printed, cancelling it will have no effect as the job is finished anyway.
- (2) If you cancel the pseudodevice name *before* releasing the job to the processing queue, the job is cancelled and you are not charged.
- (3) If you cancel a print job after it is released but before it starts printing, it will not be printed. Note: Your session charges will include the printing charges, but these will be automatically refunded at a later time.
- (4) If your print job was in the middle of printing when you cancelled it, it will stop printing at that point and you will be charged for the portion that was printed.
- (5) If you cancel a batch job before it starts executing, you will not be charged for it.
- (6) If a batch job is executing, or has already executed but has not yet printed, cancelling the job will not cancel the printout. In order to cancel the print job you must issue a second \$CANCEL. In any event you will still be charged for the job up to the point where it was cancelled.

Some typical examples:

(1) You have just copied 400 pages to the printer, and suddenly realize that it was the wrong file. The \$SYSTEMSTATUS command indicates that your job (138475) is still waiting to print, so you can use:

\$SYSTEMSTATUS 138475
1ABC:RM138475 awaiting print, P8, after 10 jobs
\$CANCEL 138475
Job 1ABC:RM138475 cancelled.

The system response gives the user's userID (in this example, 1ABC) before the job name. The job will not be printed and you will be automatically refunded for it.

(2) You have submitted a batch job (RM112984), which has already started to execute, and then you realize that you have used the wrong format:

\$CANCEL RM112984
Job 1XYZ:RM112984 cancelled.
\$CANCEL RM112984
Job 1XYZ:RM112984
Job 1XYZ:RM112984 (112985) cancelled.

When the job began execution, it created a print stream (112985) that was not cancelled the first time. The second \$CANCEL command got rid of it.

(3) After controlling *PRINT* so that it will print in portrait mode, you copy your job to it. However you do not know whether it should be printed onesided, so you decide to cancel it for now: \$CANCEL *PRINT*
PRINT RM198765 cancelled.

(The job name RM198765 was assigned by the \$CONTROL command.) The *print* cost for images and sheets will be refunded, but you will still incur charges for CPU time used by the \$LIST or \$COPY command.

(4) You have submitted a batch job to run overnight and you decide you want to cancel it, but you do not know the job name and number. You obtain it as follows:

\$SYSTEMSTATUS QUEUE *
1ABC:RM116982 awaiting execution, P10, after 35 jobs.

and now you enter:

\$CANCEL 116982
Job 1ABC:RM116982 cancelled.

Your batch job has now been cancelled.

Notes:

- (1) Use the \$SYSTEMSTATUS command to find the job names and current status of any jobs you have submitted within the last 8 hours.
- (2) Be sure you specify the correct job name when there is more than one waiting in the queue. If you have assigned the same job name to more than one job, cancelling that job name will cancel *all* jobs with that name. You can avoid this problem by cancelling the job number instead.

4.2 THE \$CONTROL COMMAND

The \$CONTROL command lets you modify the attributes of a single print or batch job. You will notice that many of the options are identical to \$SET options. The essential difference between \$CONTROL and \$SET is that \$CONTROL changes options for a particular job—only until the job is released or you sign off. The \$SET command will affect all jobs until the option is reset or you sign off. When you need a particular set of options for a single job, \$CONTROL is the command to use. It is frequently used to change the way output is printed on the Xerox 9700 page printer.

The general form of the \$CONTROL command is:

\$CONTROL *pdn* keyword

or

\$CONTROL *pdn* keyword=option

where $*pdn^*$ is a pseudodevice name from the list given below, and *keyword* (with *option* where applicable) is selected from Table 3 below. The following is a list of predefined pseudodevice names that are processed by the Resource Manager.

Pseudodevice	Facility Provided
BATCH	For submitting a batch job.
PRINT	For printing output.
PUNCH	For producing 80-byte records for BITNET files.
IMPORT	For retrieving BITNET files that have been received from another system.
EXPORT	For sending BITNET files to another system.

The first three pseudodevices have been described in other sections of this document (e.g., *BATCH* was explained in the section "Batch Processing", *PRINT* was explained in the section "Getting Printed Output", etc). Each of these sections also contains more examples of using \$CONTROL for each particular pseudodevice.

More information about BITNET is available in the document BITNET on MTS, Reference R1039.

Table 3: \$CONTROL Keywords and Options

Description

Keyword

ADDRESS="line1;line2;..."

Specify the campus mail address for delivered output (when DELIVERY=MAIL is set). This option applies only to page-printer output, not to line-printer or local-printer output.

- CANCEL *...* Cancel the job that is currently held for the specified pseudodevice.
- CLASS=char A file class to associate with the job. If no file class is assigned to a file, a class of A will be used; otherwise, CLASS is a single letter from A to Z. For BITNET connections only.
- COMMENT="text" Specify a comment that will be associated with the job. This comment will be printed on the head sheet of jobs submitted to *PRINT* or *BATCH*.
- COPIES=n Specify number of copies of output for the given pseudodevice. The default is 1.

DELIVERY={station | MAIL | NONE} Specify a delivery station to which output should be delivered by messenger service (see the file *DELIVERY for further details). If MAIL is specified, the output will be delivered by campus mail or the US Postal Service. The default is NONE.

DESTINATION=destination A BITNET destination, (e.g., userID@site, or proute@site, or croute@site).

ENCODING={NONE | NETDATA | DISKDUMP}

Specify how the data should be encoded before a BITNET transfer. By default no encoding is used. MTS files are always sent as variable-length-record files.

FILE={filename | "file name"}

A file name to associate with the job. If the file name contains blanks or

special characters, it must be enclosed in quotes. For *IMPORT*, this option can be used to select from a number of incoming jobs. For BITNET connections only.

FORMAT={LANDSCAPE | PORTRAIT | TWOUP | format-name}

Specify format for page-printer output (see *Using the Xerox 9700 Page Printer*, Reference R1038, for a complete description and list of formats). The default is LANDSCAPE. Applies only to *PRINT* and *BATCH*.

HOLD Explicitly hold the job directed to the specified pseudodevice.

JOBNAME={jobname | DEFAULT}

Assign a job name of 1 to 8 alphanumeric characters to the job directed to the specified pseudodevice. The first character must be a letter. For *IMPORT*, this option can be used to select from a number of incoming jobs. DEFAULT specifies the default format of "RM" plus six digits.

KEEPSimilar to CANCEL, but the job is requeued rather than abandoned.
Applies only to *IMPORT* for BITNET connections.

{LANDSCAPE | PORTRAIT | TWOUP}

Specify orientation of page-printer output (synonymous with FORMAT for the corresponding values). The default is LANDSCAPE. Applies only to *PRINT* and *BATCH*.

MARGIN={n.nn | NO}

Set the left margin to "n.nn" inches. "n.nn" must be less than the current page width (8.5 for portrait orientation, 11.0 for landscape). MARGIN=NO turns off the margin override and resets the margins to the default for the current format (0.5 for PORTRAIT and 0.65 for LANDSCAPE). Applies only to *PRINT* and *BATCH*.

NUMBER={(b,l,c) | NO}

NUMBER=(b,l,c) numbers pages automatically, starting with number "b", printing the number on line "l", ending in column "c". NUMBER=NO (the default) disables automatic page-numbering. The page number is always printed in the first font of the current format or FONTLIST specification.

OVERLAY={NONE | SHADED | LINED}

Specify an overlay for page-printer output (see *Using the Xerox 9700 Page Printer*, Reference R1038, for a complete description of overlay options). The default is NONE. Applies only to *PRINT* and *BATCH*.

PAGES=n Specify a page limit for a print job. The default is no page limit. Applies only to *PRINT*.

PAPER={PLAIN | 3HOLE | LABEL24 | LABEL33}

Specify paper type for Xerox 9700 page-printer output. The default is PLAIN. The obsolete ANY option is synonymous with PLAIN. Applies only to *PRINT* and *BATCH*.

PRINTER={PAGE | LINE}

Specify the type of printer. The default is PAGE. Applies only to *PRINT* and *BATCH*.

- RELEASE Release for processing the job that is currently held for the specified pseudodevice (see also the \$RELEASE command).
- ROUTE=station Specify a station for output at which output is to be printed (use the command \$COPY *ROUTE to obtain a list of valid station codes).

SEPCOPY={YES | NO}

Specify whether each copy will have separate head and tail sheets (YES) or whether all copies will be printed together as one job with a single head and tail sheet (NO). The option is only effective for page-printer output and if COPIES=n is specified. If each copy is more than 50 pages, then SEPCOPY=YES is forced. *Currently the default is SEPCOPY=YES*. On December 2, 1991 the default will be changed to NO.

SHIFT={YES | NO}Specify whether page-printer output is shifted away from the binding edge. The default is SHIFT=NO. Applies only to *PRINT* and *BATCH*.

TWOSIDED={YES | NO}

Specify printing on one or both sides of the paper for page-printer output. The default is YES. Applies only to *PRINT* and *BATCH*.

{TWOSIDED | ONESIDED}

Specify printing on one or both sides of the paper for page-printer output. The default is TWOSIDED. Applies only to *PRINT* and *BATCH*.

TYPE={PUNCH | PRINT}

Produce PUNCH- or PRINT-type files. The default is PUNCH. Applies only to *EXPORT* for BITNET connections.

Some typical examples:

(1) The next job you send to *BATCH* is to be printed onesided on 3-hole punched paper. The file called BATRUN should contain MTS commands for a batch job, including a \$SIGNON command.

The job is assigned a number and is automatically held until the \$RELEASE command is given. This demonstrates how more than one attribute may be specified on a \$CONTROL command.

(2) You want to direct the next printout to the Xerox 9700 page printer at the Michigan Union Station (UNYN):

```
# $CONTROL *PRINT* ROUTE=UNYN
 *PRINT* assigned job number 654321
 *PRINT* RM654321 held
# $LIST DATAFILE ON *PRINT*
# $RELEASE *PRINT*
 *PRINT* RM654321 released UNYN 9 images 5 sheets RATE=NORMAL
```

(3) You want your output printed with portrait orientation and a shaded overlay:

A complete list of formats and overlays is available in *Using the Xerox 9700 Page Printer*, Reference R1038.

Notes:

- (1) If a held job is not explicitly released, the keywords given on the \$CONTROL command remain in effect for all output subsequently directed to that job during the current session.
- (2) If you enter additional \$CONTROL commands for a job that is already held, the effect is cumulative. In the event of a conflict of options for a specific keyword, the last value assigned will apply to the whole job (including any output that may have been generated before the last \$CONTROL command).
- (3) You can use the \$CONTROL command with user-defined pseudodevice names.
- (4) You can use \$DISPLAY to see what is currently held for a particular job.
- (5) You can use \$CANCEL to cancel a job that is being held.

4.3 THE \$CREATE COMMAND

The \$CREATE command is normally used for creating files, but it also lets you define your own pseudodevice name with the attributes that you most commonly use (pseudodevices were explained in the "Introduction" to this document). You should consider creating your own pseudodevice if you find that you are using the \$CONTROL command frequently to modify the attributes of:

PRINT *BATCH* *PUNCH* *IMPORT* *EXPORT*

With \$CREATE you can define several print pseudodevices so you can easily produce print jobs each with a different set of predefined attributes.

Furthermore, you can use the \$CONTROL command to alter or add to the attributes of a user-defined pseudodevice. In this case, the pseudodevice is assigned a job number and held until it is explicitly released.

The general form of the \$CREATE command for pseudodevices is:

\$CREATE *pdn* TYPE=device OPEN="keywords"

where

pdn	is an arbitrary pseudodevice name, consisting of 3 to 16 characters. The first and last characters must be asterisks (*), for example, *PRINT*.
device	is one of: PRINT, BATCH, PUNCH, EXPORT, or IMPORT. This represents the type of device that is to be modified.
keywords	is one (or more) of the keywords given in Table 3 above, including any <i>options</i> that are appropriate. These attributes are applied as if a \$CONTROL command were given each time the device is opened for processing.

Some typical examples:

(1) You frequently produce notices that are printed with the attributes PORTRAIT and PAPER=PLAIN. It would be more convenient if you did not have to control *PRINT* every time you printed one, so you define a new pseudodevice name with those attributes:

\$CREATE *NOTE* TYPE=PRINT OPEN="PORTRAIT PAPER=PLAIN"

Now, any time you want to copy a notice to the printer, simply specify *NOTE* instead of *PRINT*:

This is especially useful when created in a sigfile.

(2) You have already defined the pseudodevice name *NOTE*, but you want to copy one notice with a *shaded overlay*:

The pseudodevice *NOTE* now has the attributes of portrait orientation, plain paper, and a framed overlay. This example illustrates that when you alter the attributes of a pseudodevice that you have created the attributes are *added* to those already present for that device.

Notes:

- (1) You cannot create a pseudodevice name that is already defined.
- (2) Use the \$FILESTATUS command to see what pseudodevice names are currently defined:

\$FILESTATUS *?*

This will list all the names (including predefined ones), but none of the attributes.

- (3) A user-defined pseudodevice is like a temporary file: it exists for the current session only. You can use \$CREATE in your sigfile to define one or more pseudodevice names automatically at the beginning of each session.
- (4) You can remove a user-defined device by using the \$DESTROY command.

4.4 THE \$DISPLAY COMMAND

The \$DISPLAY command lets you display options that have been set with the \$SET command, or establish whether a given pseudodevice name is defined or active.

A complete description of the \$DISPLAY command (see *MTS Volume 1: The Michigan Terminal System*, Reference R1001), shows many items that can be displayed. Only those items that specifically relate to Resource Manager job handling are included here.

The general form of the command is:

\$DISPLAY keyword

or

\$DISPLAY *pdn*

where *keyword* is one of the \$SET command keywords, and **pdn** is a pseudodevice name.

4.4.1 Displaying \$SET Command Keywords

Any RM keyword that may be used on the \$SET command may also be used on the \$DISPLAY command to determine its current setting. For example, if you had used

\$SET FORMAT=PORTRAIT

in your sigfile, you could display this information at any time during the current session:

\$DISPLAY FORMAT # FORMAT=PORTRAIT

If the item you display is set to the default value, the system would tell you that the default is in effect:

\$DISPLAY OVERLAY
OVERLAY is defaulted.

See Table 4 later in this section for a list of keywords and their default settings.

4.4.2 Displaying Pseudodevice Names

If you have started a job for a particular pseudodevice but not yet released it (for example, by using the \$CONTROL command), you can display the job name, and other information for the given pseudodevice name. For example:

\$DISPLAY *PRINT*
PRINT RM123456 routed to CNTR 4 images 2 sheets RATE=NORMAL.

You can see that you have a print job with 4 images on 2 sheets waiting to be released to CNTR at normal rates.

If you want to see which pseudodevices are currently being held you can use:

```
# $DISPLAY *?*
# *NOTE* RM113441 routed to UNYN 6 images 3 sheets RATE=NORMAL
# *PRINT* RM113440 routed to UNYN 4 images 4 sheets RATE=NORMAL
FORMAT=PORTRAIT TWOSIDED=YES
```

You currently have two jobs for the Xerox 9700 page printer at UNYN that are waiting to be released.

You may display both predefined and user-defined pseudodevice names. If the specified pseudodevice is currently active, the following information is displayed for the given device type:

Device Type	Information Displayed
PRINT	Job name and number, print route, delivery route (if any), number of images and sheets, charge rate, number of copies (if more than one), and format parameters (if any).
BATCH	Job name and number, time estimate (if any), delivery route (if any), host, print route, and format parameters (if any).
PUNCH	Job name and number, delivery route (if any), number of records, and charge rate.

Notes:

(1) If the specified pseudodevice name is defined but is not currently active for your userID, the system sends an appropriate response:

\$DISPLAY *PRINT*
PRINT is defined but is not active.

(2) You can display your current session statistics using the command:

\$DISPLAY TAILSHEET

4.5 THE \$LOG COMMAND

This command can be used to keep a record of input and output operations on a file or device. A popular use of this command is for recording your terminal session in a file. When you are logging your terminal session, the I/O operations are the commands that you are typing and the device is *MSOURCE* (the keyboard that you are using). It is useful to keep a record of your terminal session in the event that an unexpected error occurs and you need a *hardcopy* of the sequence of commands that were entered.

The general form of the \$LOG command is:

\$LOG fdname1 ON fdname2 options

where *fdname1* and *fdname2* may be files or devices and *fdname1* will default to *MSOURCE*, but *fdname2* must be specified. Options can be one of:

READS WRITES CONTROL BINARY ALL ASIS SYMBOLIC

The first five options determine which operations are logged, the last two determine the format of the \$LOG output. If you do not specify options they will automatically default to READS, WRITES, and ASIS. These options are more fully explained in the description of the \$LOG command in *MTS Volume 1: The Michigan Terminal System*, Reference R1001.

Some typical examples:

(1) You want to keep a record of your terminal session in the file LOGFILE.

\$LOG *MSOURCE* on LOGFILE

The file LOGFILE will contain all commands entered from your keyboard and all the output sent to your terminal.

(2) You are submitting a job to batch and you want to keep a record of your session in a file instead of having it printed. The commands for your batch run are in a file that looks like this:

```
$SIGNON *
$EMPTY LOGFILE
$LOG *MSINK* ON LOGFILE
$RUN SPSS:xxx INPUT=DATA PRINT=MYPRINT
$CANCEL *PRINT*
$SIGNOFF
```

Our file contains six lines. First you have your \$SIGNON command, next you \$EMPTY the file that you are using to log your session in. Note that you must \$CREATE LOGFILE before you submit this run. You cannot use a temporary file as in the previous example since you are logging in batch and it would disappear once the job was

completed. The next command instructs the system to log your job in LOGFILE. You then enter your commands for this run. In this example you are running the statistical package SPSS:xxx. Finally, you enter \$CANCEL *PRINT* which will cancel all hardcopy output from this job. Because \$CANCEL will cancel all printouts for your job, you should use it with caution. In this instance the only hardcopy that you are expecting is the log of your batch run which you are storing in a permanent file anyway.

Notes:

- (1) If you are logging terminal I/O and you \$LIST the logfile or \$EDIT the logfile and issue a PRINT /F command, the list is (potentially at least) infinite, since each line that is listed or printed is also added to the end of the logfile. In general, when using \$LIST you should specify the line range to avoid this problem, or alternatively, you can turn logging off first (see below).
- (2) You can use the \$DISPLAY command to see which files or devices are currently being logged:

\$DISPLAY LOGSTATUS
Logging *MSINK* on LOGFILE
30 lines read
192 lines written

In this example you are informed that you are logging your terminal session on the file –LOG. You have read 30 lines which have resulted in 192 lines of output into the file –LOG.

(3) You can stop logging by issuing the \$LOG OFF command:

\$LOG OFF
Logging of *MSINK* on LOGFILE terminated

4.6 THE \$RELEASE COMMAND

A job will be held if one of the following conditions apply:

- (1) You have issued a \$SET AUTOHOLD=ON command.
- (2) You have issued a \$CONTROL *pdn* for a particular pseudodevice.

In all of the above situations, the job is held until you issue a \$RELEASE command. If you neglect to use \$RELEASE, the system will do so automatically when you sign off. However, until the job is released (one way or another), it does not get submitted to the processing queue.

The general form of the \$RELEASE command is:

\$RELEASE *pdn*

When a job is released, the job name and number and attributes applied using either \$SET or \$CONTROL commands are displayed.

Some typical examples:

(1) You want to copy a file to *PRINT* and print it in portrait format:

(2) You have given a \$CONTROL command, but changed your mind before copying anything to the printer:

```
# $CONTROL *PRINT* PAPER=3HOLE
 *PRINT* assigned job number 124680
 *PRINT* RM124680 held
# $RELEASE *PRINT*
 *PRINT* RM124680 cancelled: no lines.
```

Nothing is printed, and the job is abandoned.

Notes:

- (1) If you have used \$SET AUTOHOLD=ON, you will need to explicitly release any job sent to any pseudodevice (e.g., *PRINT*, *BATCH*), regardless of whether you used a \$CONTROL command. The \$SET AUTOHOLD=ON is a useful command to put into your sigfile because it explicitly holds any files sent to a *pdn* and thus gives you the option of cancelling a job before it is released.
- (2) If you did not copy anything to the printer before releasing it, the job would be automatically cancelled (see example 2).
- (3) If there was no print job active, the system would respond that the given pseudodevice name *was not held*.
- (4) If you neglect to \$RELEASE a job, the system will do so automatically when you sign off from the current session.
- (5) Once you \$RELEASE your job the options for that pseudodevice are reset back to their default values; with the exception, of course, of those that have been changed using \$SET.
- (6) You can use \$DISPLAY to see what is currently held, for a particular device:

\$DISPLAY *PRINT*

or for all devices:

\$DISPLAY *?*

(7) If you have a number of pseudodevices active, you can release them all at once by typing the following:

\$RELEASE *?*

4.7 THE \$SET COMMAND

The \$SET command lets you set various *global* conditions. Any option specified on a \$SET command remains in effect until the end of the current session, or until it is reset to a new value. Many of the options described here apply to jobs that are directed to a printer. Some of them are valid only for the Xerox 9700 page printer.

The \$SET command is particularly useful for options that you use all the time. In this case you can put your \$SET commands into a special file called a sigfile and then have that file executed each time you \$SIGNON.

The general form of the \$SET command is:

\$SET option

where option is selected from Table 4 below.

Note: Only the options that specifically relate to RM job handling are included here; for complete documentation of the \$SET command, see *MTS Volume 1: The Michigan Terminal System*, Reference R1001.

Table 4: \$SET Options

Option

Description

ADDRESS="line1;line2;..."

Specify the campus mail address for delivered output (when DELIVERY=MAIL is set). This option applies only to page-printer output, not to line-printer or local-printer output.

AUTOHOLD={OFF | ON}

Specify that all subsequent jobs sent to *BATCH*, *PRINT*, and *PUNCH* will automatically be held until explicitly released. This will also apply to all user-defined pseudodevices. The default is OFF.

COMMENT="text" Specify a comment to be printed on the head sheet of the job.

COPIES=n Specify number of copies of output for print jobs. The default is 1.

DELIVERY={station | MAIL | NONE}

Specify a delivery station to which output should be delivered by messenger service (see the file *DELIVERY for further details). If MAIL is specified, the output will be delivered by campus mail or the US Postal Service. The default is NONE.

FORMAT={LANDSCAPE | PORTRAIT | TWOUP | format-name}

Specify format for page-printer output (see *Using the Xerox 9700 Page Printer*, Reference R1038, for a complete description and list of formats). The default is LANDSCAPE.

JOBNAME={jobname | DEFAULT}

Specify a job name of 1 to 8 alphanumeric characters for all subsequent jobs sent to the printer. The first character must be a letter. DEFAULT specifies the default format of "RM" plus six digits.

MARGIN={n.nn | NO}

Set the left margin to "n.nn" inches. "n.nn" must be less than the current page width (8.5 for portrait orientation, 11.0 for landscape). MARGIN=NO turns off the margin override and resets the margins to the default for the current format (0.5 for PORTRAIT and 0.65 for LANDSCAPE).

NUMBER={(b,l,c) | NO}

NUMBER=(b,l,c) numbers pages automatically, starting with number "b", printing the number on line "l", ending in column "c". NUMBER=NO (the default) disables automatic page-numbering. The page number is always printed in the first font of the current format or FONTLIST specification.

OVERLAY={NONE | SHADED | LINED}

Specify an overlay for page-printer output (see *Using the Xerox 9700 Page Printer*, Reference R1038, for a complete description of overlay options). The default is NONE.

PAGES=n Specify a page limit for a print job. The default is no page limit. It applies only to jobs directed to *PRINT* or user-defined pseudodevices with TYPE=PRINT.

PAPER={PLAIN | 3HOLE | LABEL24 | LABEL33}

Specify paper type for Xerox 9700 page-printer output. The default is PLAIN. The obsolete ANY option is synonymous with PLAIN.

PRINTER={PAGE | LINE}

Specify the type of printer. The default is PAGE.

ROUTE=station Specify a station for output (use the command \$COPY *ROUTE to obtain a list of valid station codes).

SEPCOPY={YES | NO}

Specify whether each copy will have separate head and tail sheets (YES) or whether all copies will be printed together as one job with a single head and tail sheet (NO). The option is only effective for page-printer output and if COPIES=n is specified. If each copy is greater than 50 pages, the SEPCOPY=YES is forced. *Currently the default is SEPCOPY=YES*. On December 2, 1991 the default will be changed to NO.

SHIFT={YES | NO}Specify whether page-printer output is shifted away from the binding edge. The default is SHIFT=NO.

TWOSIDED={YES | NO}

Specify printing on one or both sides of the paper for page-printer output. The default is YES.

Note: The above list gives Resource Manager options that you might want to set globally. For a complete list of \$SET options, refer to *MTS Volume 1: The Michigan Terminal System*, Reference R1001.

Some typical examples:

(1) You have decided that *all* output directed from the current session to *PRINT* should be printed in portrait format, on one side only:

\$SET FORMAT=PORTRAIT TWOSIDED=NO

All subsequent output for the Xerox 9700 page printers will be printed in portrait mode, on one side of the paper. This demonstrates how more than one option may be used on a \$SET command.

(2) You are working in the Church Street (CHUR) computing site and you want to print all your output for the current session to the the Xerox 4045 printer located there:

\$SET ROUTE=CHUR

Anything sent to *PRINT* will now be printed at that computing site. Remember that this type of printer cannot handle many of the characters and specifications that are available only on the Xerox 9700 page printers.

(3) You want to ensure that none of your print jobs exceed 20 pages of output so you enter:

\$SET PAGES=20 AUTOHOLD=ON

Now whenever you \$COPY or \$LIST a file to *PRINT*, only 20 pages (images) will be copyed after which an error message:

PRINT assigned job number 348902
***Local Page Limit Exceeded

will be printed. This option can be overidden for a particular print job by using the \$CONTROL command.

Notes:

- (1) The \$SET options remain in effect for the whole session, or until explicitly reset.
- (2) Using \$SET does not initiate any particular "job".
- (3) Values that have been set may be displayed (see the \$DISPLAY command).
- (4) If you submit a batch job from a terminal session or from another batch job, the attributes of the job will be defaulted from any values that have been \$SET in the submitting job unless overridden by the \$SIGNON command in the submitted job.

4.8 THE \$SYSTEMSTATUS COMMAND

The \$SYSTEMSTATUS command lets you find out the current status of any job (or jobs) that you have submitted for processing within the last 8 hours, or that has not finished processing. It can also tell you how busy the computer is overall.

The general form of the command is:

\$SYSTEMSTATUS USERS

or

\$SYSTEMSTATUS QUEUE option

where *option* selects the job(s) for which you want queuing information.

4.8.1 The USERS Keyword

This form of the command gives you a general idea of the system load. The information displayed includes:

- (1) number of terminal users signed on
- (2) number of active batch jobs
- (3) number of active non-MTS jobs
- (4) number of virtual pages in use
- (5) number of real pages in use

4.8.2 The QUEUE Keyword

This displays the status of individual jobs that you have submitted for processing. It also provides summary information about queue lengths. You must be signed on to the userID that submitted the job in order to see its current status.

The general form of the command is shown above, and the available options are listed below.

Option	Information Displayed
jobnumber	The current status of the specified job.
jobname	The current status of the specified job.
USER	The status of all jobs submitted by the current userID, but not yet processed.
*	The status of all jobs processed for the current userID during the preceding 8-hour period. This will include jobs that have printed, or were cancelled, or are still waiting to be processed. This will also include BITNET files that are being held in *IMPORT*.

ALL	A summary including the number of active and queued jobs, the number of pages waiting to print for all printers, and the number of plot minutes in the queue. This information represents the whole system, not just the current userID.
PROUTE=code	A summary of all jobs awaiting print at a specified print station.
IMPORT	A summary of all BITNET files that are awaiting import.

Some typical examples:

(1) If you just want to get an idea how busy the system is:

```
# $SYSTEMSTATUS USERS
There are 267 terminal users, 2 batch jobs, 216 available
lines, and 60 non-MTS jobs using 20150 private virtual pages
and 9110 real pages. Additionally, there are 1270 shared
virtual pages.
```

(2) The owner of the userID 1ABC submitted two jobs and wants to see if they have finished printing.

```
# $SYSTEMSTATUS QUEUE USER
1ABC:RM116982 is awaiting print, P14, after 0 jobs.
```

This tells you that one job is next in the queue to be printed. In order to see the status of all jobs (included those that have printed within the last 8 hours) you should enter:

```
# $SYSTEMSTATUS QUEUE *
1ABC:RM116982 is awaiting print, P14, after 6 jobs.
1ABC:RM118973 is awaiting purge, was printed on PTR2 at
10:33:00 Mon Jan 15/90.
1ABC:RM113243 was printed on PTR2 at 09:24:11 Mon Jan 15/90.
```

The first job is awaiting print and there are six jobs ahead of it in the queue. The second entry represents a job that has been printed but has not yet been *purged*. For an explanation of the the term "purged", see "When is a Job Purged?" in this section. The last entry represents a job that has been printed and has also been purged.

- (3) How many jobs are waiting to print on the Church Street (CHUR) computing site printer?
 - # \$SYSTEMSTATUS QUEUE PROUTE=CHUR
 1 active job, 3 queued jobs, representing 81 pages.
- (4) You want to track a job that was submitted on the UB-MTS system and tranferred to the UM-MTS system for printing. From the UB-MTS system, you can enter

```
# $SYSTEMSTATUS QUEUE RM123456
1ABC:RM123456 is awaiting purge, was sent at 12:10:15
Mon Jan 15/90.
```

This shows the time the job was sent to UM-MTS for printing. If you have an account with the same userID on the UM-MTS system, you can enter

```
# $SYSTEMSTATUS QUEUE RM123456
1ABC:RM123456 (345678) is awaiting purge, was printed on PTR2
```

at 12:20:20 Mon Jan 15/90.

Note that the UB-MTS job (numbered 123456) was given a new job number of 345678 when it was transferred to the UM-MTS system. However, the job name RM123456 remained the same.

4.8.3 Explanation of Responses

The following list summarizes the responses you can normally expect when querying the status of a job.

(1) job name is awaiting execution, Pm, after n jobs

This indicates a batch job that is waiting in the execute queue. It also tells you the priority of the job, and the number of batch jobs ahead of it. (For more information on priorities, see the section "Batch Processing".)

(2) *job name* is executing

This indicates a batch job that is currently executing.

(3) *job name* is awaiting print, Pm, after *n* jobs

This indicates a job that is waiting in the print queue. It also tells you the priority of the job, and the number of print jobs ahead of it.

(4) *job name* is printing

This indicates a job that is currently printing.

(5) *job name* is awaiting purge, was printed on PTRn at ...

This message gives the time and date at which the job was printed. The job is finished, but has yet to be purged. (See Note 3 at the end of this section.)

(6) *job name* was executed at ...

This indicates the job has been executed.

(7) *job name* was printed on PTRn at ...

This indicates the job was printed and has been purged from the system.

(8) *job name* was cancelled at ...

This indicates the job was cancelled (see the \$CANCEL command).

(9) *job name* not locatable

This indicates you are requesting information for a job that the RM does not have information for. Perhaps it finished more than eight hours ago, or was submitted under

another userID.

(10) *job name* was sent at ...

This indicates the job was sent to another system for processing. This message applies to print, punch, or BITNET jobs that require a device that is attached to another system. All print jobs from the UB-MTS system are sent to the UM-MTS system for printing.

Notes:

- (1) \$SYSTEMSTATUS always looks for jobs for the current userID. It automatically prefixes each job name with this userID.
- (2) Note the difference between the following QUEUE options:

```
# $SYSTEMSTATUS QUEUE USER
No jobs found for ID 1ABC.
# $SYSTEMSTATUS QUEUE *
1ABC:RM119435 was cancelled at 15:36:23 Wed Jan 10/90.
```

The * will find jobs that were purged or cancelled earlier in the day, whereas "USER" locates only those that are active or pending.

(3) The Xerox 9700 page printer maintains its own internal print queue. The time given as printed is actually the time the Resource Manager copied the job to the printer. If there is a heavy backlog inside the printer or it is out of paper, there may be a lag between the time given and the time the job really gets printed.

4.8.4 When is a Job Purged?

The expression "awaiting purge" simply means that the job is finished, but still exists on the system. So, for example, if operators discovered your output was printed while the Xerox 9700 page printer was malfunctioning, they could reprint the job when the problem was fixed. This saves you the task of rerunning the job.

After a set amount of time, the Resource Manager purges jobs which have already printed in order to make room for new ones being submitted. The length of time a job remains in the purge queue varies depending on the type of job.

4.9 THE \$VIEW COMMAND

The VIEW command allows users to look at batch and print jobs that are queued by the Resource Manager for execution or printing.

For example, you can look at the contents of the print job 123456 by first entering the command

VIEW 123456

at the MTS "#" prompt. Then, you can use MTS File Editor commands (including visual mode) to examine the job. You can copy sections of the job to other files, cancel it, or reroute it. If the job is queued for execution, you will see the commands that make up the batch job. If the job is queued for

printing, you will see the output produced by the job.

Print jobs must be released to the Resource Manager before they can be viewed. That is, you cannot give the commands

COPY file *PRINT*
> *PRINT* assigned job number 123456
VIEW 123456

but you can give

```
# COPY file *PRINT*
> *PRINT* assigned job number 123456
# RELEASE *PRINT*
 *PRINT* RM123456 released to ...
# VIEW 123456
```

You can view print output only while it is queued for printing. Once printed, it can no longer be viewed.

You can send jobs to the VIEW command by using the ROUTE=VIEW option on the \$CONTROL or \$SIGNON commands. Jobs submitted with this routing will not print but will remain queued in the system for up to 11 days until rerouted or canceled (both described later).

For example, to use ROUTE=VIEW for a print job, use the commands

CONTROL *PRINT* ROUTE=VIEW COPY file *PRINT* RELEASE *PRINT*

You can then use the VIEW commands described below. For batch jobs, place the parameter ROUTE=VIEW on your \$SIGNON command.

If you currently submit batch jobs and only examine the printed output to determine the success or failure of the job, then VIEW can be of use to you. If you submit your job with a ROUTE=VIEW, you can examine the output from your terminal. Having looked at it, you can then cancel it.

If you submit a batch job that is going to generate a lot of output, it might be worth first routing the output to VIEW. You can then examine the output and, if the job is correct, reroute it to the printer.

The VIEW command requires a job number to identify the job. You can find the job number by using the LOCATE command. To see what jobs you have in the system, enter the command:

LOCATE

or, to see only the jobs with ROUTE=VIEW:

LOCATE VIEW

You will see a message such as

RM299444 (234567) waiting print, posn 0, Route=VIEW. RM299593 (234568) waiting print, posn 1, Route=VIEW.

50 Managing Batch and Print Jobs

Then, you can use the VIEW command to view a job:

VIEW 234567

You will see a message such as

* Print job 234567 RM234567 1ABC 3 pages submitted at 11:34:32 Tue Jul 30/91

The asterisk "*" is a prompt that indicates the VIEW command is active. To examine the job in visual mode, enter the following command at the "*" prompt:

v

Subsequently, you could cancel the job with the command

CANCEL

To exit the VIEW command, enter

STOP

at the "*" prompt. For more help on the VIEW command, enter

HELP

while you are in the VIEW command.

You can reroute jobs from VIEW to a printer by entering the following command at the "*" prompt:

ROUTE station

For example,

ROUTE CNTR

will route the output of a job to the CNTR batch station. You can send jobs to other printers, such as the page printer at NUBS, by substituting NUBS for CNTR. Please note that you will probably not want to view text-formatted jobs, such as .DVI files, since their output normally is not readable using the File Editor.

If you work under more than one MTS userID, please note that you can only view jobs from the userID on which you originally executed them.

5. THE *RMLOG PROGRAM

The Resource Manager maintains a set of log files that contain a recent history of all jobs submitted to the system for processing, including print jobs, batch jobs, and BITNET jobs such as file transfers and messages. These log files contain recent history data from the spooling system. Currently you can view 14 days worth of history data using the log files. This data is currently updated every one-half hour.

You may use the *RMLOG program to track the processing of your job if it was submitted more than one-half hour ago. The program is invoked with the command

\$RUN *RMLOG [INPUT=input] [SERCOM=error-messages]

The input read from INPUT (SCARDS) consists of an arbitrary number of commands, one per line. A command is one of the following:

\$MTS-command	Issues the given MTS command.
*comment	Does nothing.
FORMAT {FREE	FIXED} Sets the format for printed output.
HELP topic	Prints help text for the given topic.
MTS	Returns to MTS (*RMLOG can be restarted).
PRINT options	Selects and prints entries from the RM log files. See the description of the print options below.
STOP	Stops the program.
TIME {OFF SHORT DATE FULL} Sets the time format.	

5.1 PRINT COMMAND OPTIONS

The PRINT command is the basic information display command in *RMLOG. PRINT searches the log files for entries satisfying criteria given on the command and prints the information found. The log data includes entries for most processing steps of a job (such as a message being transmitted) and some information about the job itself (such as size, overlay, and paper specifications).

Because the log files are large, specifying good search criteria will save searching, waiting, and printing time. The files are indexed by userID, but the number of entries generated for one userID in 14 days can still be large as one job typically generates several entries. For example, when searching for the records for a job whose name or number you do not know, specify a time range whenever possible; e.g., TIME="YESTERDAY" will prevent the *RMLOG program from searching several files and printing several lines for every job processed during the last 14 days.

*RMLOG will select only entries for jobs belonging to the userID from which it is run.

The syntax of the PRINT command is

PRINT options [ON FDname]

where "options" may be any of the search criteria options specified below. If several options are entered (separated by blanks), only entries satisfying *all* of the search criteria will be printed. The ON FDname parameter allows you to redirect the output to a file or device. The print options, with the minimum abbreviations underlined, are:

DESTNODE=dest-node	Select BITNET entries with the given destination node.
DESTUSER=dest-user	Select BITNET entries with the given destination user.
DEVICE=device-name	Select entries with the given device name.
ID=origin-id	Select entries with the given four-digit origin-id. Synonym for ORGID.
JOB#=job-number	Select entries for the given job number. Synonym for NUMBER.
JOBNAME=job-name	Select entries for given job-name.
KIND=record-types	Select entries of the given record-types. See the list of record types below.
LINK=link-name	Select BITNET entries with the given link-name.
MESSAGE=msg-number	Select entries with the given message number. Synonym for MSG#.
MSG#=msg-number	Select entries with the given message number. Synonym for MESSAGE.
NUMBER=job-number	Select entries for the given job number. Synonym for JOB#.
ORIGID=origin-id	Select BITNET entries with the given four-digit origin-id. Synonym for ID.
ORIGNODE=origin-node	Select BITNET entries with the given origin-node.
PROCESSOR=proc-name	Select entries for the given processor.
<u>Q</u> UEUE=queue-name	Select entries for the given queue.
ROUTE=route	Select entries with the given print route.
SENTNODE=sent-node	Select BITNET "sent" messages from the given node.

SENTLINK=sent-link	Select BITNET "sent" messages for the given li	ink.
TIME="time/date-range"	Select entries within the given time/date-ran for further information on specifying time/date	
TYPE=record-types	Select entries of the given record-types. See types below.	the list of record

To access the BITNET records for a job, you must specify ORIGID=origin-id to select the records for only that BITNET job. The origin-id of a job is the last four digits of the job's Resource Manager job number. To trace job number 119353, include ORIGID=9353. See also the example below detailing the tracing of a BITNET message.

5.2 TIME/DATE RANGES

A time/date range for the TIME option on the PRINT command may be given in the form

time/date

or

time/date1 TO time/date2

Time/date input conversion is driven by a grammar that attempts to accept most styles of time/date presentation that are commonly used. It generally requires that the components specified be meaningful and unambiguous.

An input time may be composed of the following pieces. They may be put together in most meaningful combinations, using commas and blanks as conventional separators.

Year	A year is a two-digit number in the range 32 to 99, inclusive, or a four-digit number in the range 1900 to 1999, inclusive. In some unambiguous contexts, a two-digit number in the range 0 to 31 will also be recognized as a year.
Month	A month is one of "January", "February",, "December". Month names may be abbreviated to their first three characters.
Day	The day of the month is specified by a number in the range 1 to 31, inclusive.
Time	Times are integer strings containing ":" and/or "." and/or followed by "AM" or "PM" (or "A.M." or "P.M.") A fully specified time appears as "23:22:05.67835".
	The time must specify the hours and minutes, with the seconds and microseconds components optional. The minutes component may also be omitted if "AM" or "PM" appears.
	Thus, the following are all allowed:
	12 AM

23:59 00:01:35.5

Day of Week Days of the week are specified as "Monday", "Tuesday", ..., "Sunday". They may be abbreviated to the first three characters of the day name ("Mon", "Tue", etc.).

The day-name is just checked for correctness with respect to the other date components. Currently, a day-name cannot be used by itself.

- MM–DD–YY or MM/DD/YY The MM–DD–YY form of month, day and year is supported. For example, "11-06-88" is November 6, 1988.
- YY.DDD For compatibility with certain IBM software, the form YY.DDD is supported, where YY is a two- or four-digit year number and DDD is the day number in the year. For example, "89.103" is April 13, 1989.
- Other Forms The following strings are recognized, with the obvious meanings.

YESTERDAY TODAY NOW

These first two forms represent a date only, and may be combined with a time specification. The form "NOW" represents the current time (to the microsecond level).

It is not necessary to specify all the components of the time (year through microsecond). The input conversion routines will default unspecified components in many cases. For date ranges, the default time is "00:00:00.0" on the lower date and "11:59:59.9" on the upper date; thus, "Jan 18 To Jan 19" specifies two full days.

The following examples illustrate the use of time/date ranges:

```
PRINT TIME="Jan 18/1990"

PRINT TIME="Thu January 18/90"

PRINT TIME="Thu January 18/90"

PRINT TIME="Jan 18"

PRINT TIME="1/18/90

PRINT TIME="90.018"

PRINT TIME="90.018"

PRINT TIME="YESTERDAY"

PRINT TIME="January 18/1990 To Jan 19/90"

PRINT TIME="January 18/1990 To NOW"

PRINT TIME="January 18/1990 To 18:00 Jan 18/90"

PRINT TIME="6:00 PM YESTERDAY TO 6:00 AM TODAY"

PRINT TIME="11:15:20 Jan 18 To 11:45:30 Jan 18"

PRINT TIME="January"

PRINT TIME="January 1990"
```

5.3 RECORD TYPES

The record types can be either a single record type, or a parenthesized list of record types, separated by blanks or commas. Some of the more useful individual record types are:

QUEUE	A record of a job being put on a queue.
PROC	A record of a job being given to a processor.
ACC	A record of the accounting data from the completion of a job.
DONE	A record of the completion of a job.
NETJOBIN	A record of the receipt of a BITNET job.
NETJOBOUT	A record of the sending of a BITNET job.
NETMSG	A record of a BITNET network message or command.
MSGJOBIN	A record of the transformation of an incoming BITNET message to a
	message-system message.
MSGJOBOUT	A record of the transformation of an outgoing BITNET message from a
	message-system message.
RNMSENT	A record of a BITNET network message that indicates that an intermediary
	site has forwarded a job from this site onwards.

Additionally, the following aggregate abbreviations are defined:

JOB	means (QUEUE, PROC, DONE, ACC)
NETJOB	means (NETJOBIN, NETJOBOUT, RNMSENT)
MSGJOB	means (MSGJOBIN, MSGJOBOUT)
NET	means (NETJOB, MSGJOB)
SENT	means (NETJOBOUT, RNMSENT)

5.4 EXAMPLES

The following example shows the basic use of the program to obtain data on jobs that were printed yesterday. The "Log_" is a prompt printed by the program; the remainder of the line was typed by the user.

```
# run *rmlog
Log_ print time="yesterday"
Wednesday January 10, 1990
16:41 1ABC:RM193730 queued to CNTR.
16:43 1ABC:RM193733 queued to UNYN.
16:43 1ABC:RM193733 on proc PTR2.
16:43 1ABC:RM193733 on UNYN PTR2, 61 Ppp, 8,378 Ppl, 61 PPI, 31 PPS.
16:43 1ABC:RM193733 done.
16:45 1ABC:RM193730 done.
```

In the above example, Ppp means page-printer pages, Ppl means page-printer lines, PPI means page-printer images, and PPS means page-printer sheets.

The following example illustrates the tracking of outgoing message number 1352830.

```
# run *rmlog
Log_ print time="14:00 to now" msg#=1352830
Friday January 12, 1990
14:40 SYSM:RM119353 from MAILER@UBCMTSG to MAILER@UMICHUM output from
message 1352830 from Paul Whaley.
```

Log_ print time="14:00 to now" origid=9353 Friday Janaury 10, 1990 14:40 SYSM:RM119353 from MAILER@UBCMTSG to MAILER@UMICHUM output from message 1352830 from Paul Whaley. 14:40 RM119353 from MAILER@UBCMTSG to MAILER@UMICHUM sent on link SFU, 8 records. 14:40 SFU sent file 9353 from MAILER on link SFUV to MAILER@UMICHUM. 14:40 SFUV sent file 9353 from MAILER on link UALTAVM to MAILER@UMICHUM. 14:40 UALTAVM sent file 9353 from MAILER on link SASK to MAILER@UMICHUM. 14:51 SASK sent file 9353 from MAILER on link UREGINA1 to MAILER@UMICHUM. 14:57 UREGINA1 sent file 9353 from MAILER on link UOFMCCX to MAILER@UMICHUM.

The "sent file" messages indicate that the last record of the BITNET job according to the RM log files is that it is at UOFMCCX (the University of Manitoba). Note that the RM log files may be incomplete; these kinds of messages are only relayed to the originating site if all intermediate links are working, or the RM log files may not have been updated yet.

APPENDIX A GDINFO CODES AND SENSE DATA

GDINFO TYPE CODES

If you have programs that call GDINFO, you should be aware that the printer device-type code HPTR (see *MTS Volume 3, System Subroutine Descriptions*, Reference 1003) is no longer set for *PRINT*. Now, the code RMPT must be used instead. Other corresponding changes are RMRD for the input to a batch job (*MSOURCE*), RMPC for *PUNCH*, and RMBA for *BATCH*.

SENSE DATA

The sense data returned by the RM for pseudodevices differs slightly from that returned by HASP. Only the first 20 bytes are the same. See Sense_Data_Type in the file *PLUS.SOURCELIB for a definition of the new form.

BATCH, 6, 7

INDEX

EXPORT, 32 *IMPORT*, 32 *PRINT*, 21 ADDRESS option, 14, 26, 33, 43 AUTOHOLD option, 43 batch, *BATCH*, 7 advantages, 6 checking your job, 8 conditional sign-off, 19 cost of using, 10 creating a batch file, 7 definition, 6 default options, 11 errors, 10 job names, 14 job statistics, 16 output, 9, 16 queues, 9 reruning, 19 sigfiles, 20 submitting a job, 7 tapes, 14 turn-around time, 8 BITNET, 32 Cancel, batch jobs, 31 print jobs, 31 CANCEL command, 9, 18, 27, 30, 36 CANCEL option, 33 cancelling for another user, 30 character set, 23 charges, 31 CLASS option, 33 COMMENT option, 33, 43 CONTROL command, 32 COPIES option, 14, 26, 33, 43 COPY command, 21 CREATE command, 17, 27, 36

deferred rates, 12

DELIVERY option, 14, 25, 33, 43 DESTINATION option, 33 DESTROY command, 38 DISPLAY command, 24, 36, 38 pseudodevices, 39

ENCODING option, 33 execution priority, 11

FILE option, 33 FILESTATUS command, 38 FORMAT option, 14, 23, 34, 43

global limits, 10

hardcopy, 21 head sheet, 16 HOLD option, 34

job names, 4, 30 job numbers, 4, 30 job priority, 11 JOBNAME option, 14, 34, 43

KEEP option, 34

LANDSCAPE option, 34 LIST command, 21 LOG command, 9, 31, 32, 40 low rates, 12

MARGIN option, 14, 34, 44 minimum rates, 12 multiple print streams, 17

normal rates, 12 NUMBER option, 34, 44

ONESIDED option, 14, 35 OPEN option, 28 OVERLAY option, 14, 23, 34, 44

PAGES option, 10, 13, 14, 34, 44 PAPER option, 14, 34, 44 PORTRAIT option, 14, 34 print priority, 11 printed output, 32 PRINTER option, 14, 34, 44 printing, altering options, 22 cancel, 27 carriage-control, 21 control options, 23 COPY command, 21

creating a device, 27 default options, 22 delivery, 25 **DISPLAY** command, 24 large jobs, 21 lined paper, 23 LIST command, 21 multiple copies, 26 queue, 21 routes, 25 set options, 22 signon options, 22 summary of options, 25 9700 commands, 25 pseudodevices, 5, 17, 32, 36, 38 purge, 49 queue, definition, 4 RATE option, 8, 13, 14 rates, 12 refunds, 31, 40 **RELEASE** command, 41 **RELEASE** option, 35 **RERUN** option, 14 ROUTE option, 14, 25, 35, 44 SEPCOPY option, 14, 26, 35, 44 SET command, 32, 38, 43 SHIFT option, 14, 35, 44 sigfiles, 20, 22, 38 SIGNON command, 11, 13, 14, 20 SYSTEMSTATUS command, 8, 27, 32, 46 tail sheet, 16, 39 tapes, 14 TAPES option, 14 TIME option, 10, 13, 14 TWOSIDED option, 14, 35, 44 TWOUP option, 34 TYPE option, 28, 35 user-defined pseudodevices, 17, 36, 39

VIEW command, 49

WAITUNTIL option, 14

9700, 25