

CHAPTER 8

BATCH

OS/78 BATCH processing is ideally suited to lengthy or after-hour jobs consisting of fixed sequences of OS/78 system commands not requiring operator intervention. BATCH allows you to create a file that contains the OS/78 commands. Output can be line printer listings and files. These files contain program output or comprehensive summaries (log) of all action taken by the program, OS/78 BATCH, and the operator.

BATCH provides optional spooling of output files. Spooling allows printed output to be diverted to a fast file-structured device, such as a diskette, for later transferral to the slower device. This feature serves to increase perceived throughput on the system. A line printer, although optional, makes the use of BATCH more effective.

With a few exceptions, BATCH executes standard OS/78 commands.

8.1 BATCH PROCESSING UNDER OS/78

OS/78 BATCH maintains an input file and an output log. The BATCH input file consists of a series of BATCH commands. The input file must reside on the system device (disk pack or diskette). Its default extension is .BI. Each command in the BATCH input file generally occupies one line.

The BATCH output file is a line printer listing (log) on which BATCH prints job headers, certain messages that result from conditions within the input file, an image of each line in the input file, and certain types of user output. BATCH supports only the LA78 line printer for the output of the BATCH log. Listings, however, can be output on the LQP78 line printer. If a line printer is not present in the system, the output file is displayed on the terminal.

BATCH accepts program and data files from any input device in the system. User output files may be directed to any output device in the system.

BATCH also permits optional spooling of output files. When spooling is requested, every output request to a non-file-structured device output file is assigned a file name from a list of names maintained by BATCH and directed to a file-structured spool device instead of the user-specified device. Spooling of output files increases BATCH throughput, permitting slow output operations to be postponed until a more favorable time. For example, a batch processing run that generates many output listings may be initialized to reroute all listings to a specified diskette. The listings on the diskette may then be dumped onto the appropriate hard copy device after the run, when more time is available. The spool device may be any OS/78 file-structured device.

BATCH

BATCH is called via the SUBMIT command. The format for a BATCH command string is:

```
SUBMIT spool-dev:<dev:input/options
```

where:

spool-dev: is the device on which to spool output. If not specified, no spooling is performed. Note that spooling applies only to non-file-structured output.

dev:input is the input device and file. The default extension for BATCH input files is .BI. The default input device is DSK.

/options are listed in Table 8-1.

Table 8-1
BATCH Run-Time Options

Option	Meaning
/E	Treat OS/78 Monitor errors as non-fatal errors. If /E is not specified, monitor errors cause the current BATCH job to be terminated.
/H	Do not output a BATCH log.
/Q	Output an abbreviated BATCH log, consisting of \$JOB and \$MSG lines.
/T	Output the BATCH log to the terminal. This option need be specified only if a line printer is available. If a line printer is not available, the BATCH log is automatically output to the terminal.
/U	BATCH will not pause for operator response to \$MSG lines. Any attempt to use TTY: as an input device to an unattended BATCH stream will cause the current job to be aborted.

8.2 BATCH COMMANDS

A BATCH command is a character or string of characters that begins with the first character of a line in the BATCH input file. Each BATCH command must be followed by a RETURN. The files may contain form feed characters, but form feed characters are ignored by BATCH on input.

BATCH recognizes four monitor level commands. These commands allow routine housekeeping operations in a multi-job, batch processing environment and provide communication between the BATCH user and the operator. Table 8-2 lists the BATCH commands, which may be considered as an extension of the OS/78 Monitor command set. Note that the first character of the \$JOB, \$MSG and \$END commands is a dollar sign (SHIFT/4).

BATCH

Table 8-2
BATCH Commands

Command	Meaning
\$JOB	Initialize for a new job and display a job header on the output file. The remainder of the \$JOB record is included in the job header but ignored by BATCH. It should be used for job identification, to provide correlation between terminal output, line printer output and spool device output.
\$MSG text	Sound the terminal buzzer and print the message specified by the argument text at the terminal. If the /U option was not specified, implying that an operator is present, BATCH will pause until any key is struck at the keyboard. If the /U option was specified, processing continues uninterrupted.
\$END	Terminate batch processing and exit to the monitor. A \$END command should be the last line of every BATCH input file.
/	Copy the line onto the output log, then ignore it. BATCH assumes that every line beginning with a slash is a comment.

Any record that begins with a dollar sign character but is not one of the BATCH commands listed above is copied onto the output file and ignored by BATCH.

A BATCH processing job consists of a \$JOB command line and all of the commands that follow it up to the next \$JOB or \$END command. Normally, all the commands submitted by one user are processed as a single job, and all output from these commands appears under one job header.

After BATCH encounters a \$JOB command, it scans the input file until the next OS/78 command is read. Any lines that follow the \$JOB command and precede the first command are written onto the log and ignored by BATCH.

The first character of every command is a period (.). An exception to this is the use of an asterisk (*) whose function is to accept a command string that indicates input/output files. It is further described in the example program given in Section 8.3. The rest of the line contains a command, which should appear in standard OS/78 format. However, commands that would be terminated with an ESCape under interactive OS/78 should be terminated with a dollar sign under BATCH. Most OS/78 commands are legal input to BATCH except the MEMORY and SQUISH SYS: commands. ODT will go to the terminal for input instead of the BATCH file. It is not usually meaningful to invoke ODT, EDIT, or other interactive programs under BATCH.

BATCH executes an OS/78 command by stripping off the initial period (.) character and passing the remainder of the line to the monitor. BATCH then passes control to the monitor, which executes the command as though it had been typed at the keyboard. Monitor commands that return control to the monitor level should be followed by a BATCH command or another monitor command. The SUBMIT command can be used to chain from one BATCH stream to another.

BATCH

The general rules and conventions associated with BATCH processing are as follows:

1. The dollar sign (\$) is always in the first character position of the BATCH command lines \$JOB, \$MSG and \$END.
2. Each job must have a \$JOB and \$END command.
3. OS/78 commands can be spelled out entirely or in accordance with the accepted abbreviations. Also, a period (.) must precede every command.
4. Wildcards can be specified only for the OS/78 commands COPY, DELETE, DIRECT, LIST, RENAME and TYPE.
5. Comments may only be included as separate comment lines.
6. Only 80 characters per control statement are allowed.
7. Continuation file specification lines (where necessary) are specified by an asterisk at the beginning of the line:

```
.LOAD PROG,SUBA,SUBB$  
*SUBC,SUBD,SUBE$
```

8.3 THE BATCH INPUT FILE

The following file is an example of a BATCH input file.

```
$JOB  
.DATE 13-MAY-79  
/LIST ANY HELP FILES AND STARTING BLOCKS  
.DIR *.HL/B  
$MSG INSERT DISKETTE INTO DRIVE 1 - TYPE ANY KEY TO CONTINUE  
/LIST DIRECTORY OF DISKETTE 2 - SPOOL TO FILE BTCHA1  
.DIR TTY:<RXA1:  
/COPY FILE FROM DISKETTE 2 DISKETTE 1  
.COPY RXA0:<RXA1:POWER.FT  
/COMPILE FORTRAN PROGRAM POWER.FT  
.COMPILE POWER.FT  
/LOAD FORTRAN PROGRAM AND CHAIN TO RUN TIME SYSTEM  
.LOAD POWER.RL/G$  
/STORE RESULTS OF FORTRAN PROGRAM IN FILE 'HOLD'  
*HOLD.TM</4$  
/EXECUTE FORTRAN PROGRAM AND DISPLAY RESULTS ON SCREEN  
.EXE POWER.LD  
/END OF JOB NO.1 - START NEXT JOB  
$JOB  
/ASSEMBLE FILE SAMPLE AND PRODUCE CREF LISTING  
.PAL SAMPLE/C-LS  
/EXECUTE PROGRAM SAMPLE  
.EXE SAMPLE.BN  
/END OF EXAMPLE AND BATCH INPUT FILE  
$END
```

The file was created using the Editor and is named BATSAM.BI. For the example used, both the PAL8 and FORTRAN IV system programs were on diskette RXA0.

BATCH

BATCH is started using the SUBMIT command and typing

```
.SUBMIT RXA0:<BATSAM
```

when spooling is desired, or

```
.SUBMIT BATSAM
```

when spooling is not desired. The default extension for the BATCH input file is .BI.

BATCH begins processing by printing a job header and executing the DATE command. BATCH next executes the DIRECT command, which displays any Help file names and their starting blocks. Wildcards are used in this command.

BATCH continues to scan the input file for the next line. BATCH processes the \$MSG command, sounds the terminal buzzer, and copies the \$MSG record onto the terminal. Assuming that an operator is present, processing is suspended until any key is typed at the terminal.

The operator performs the task specified by the \$MSG command (that of placing diskette 2 into Drive 1) and types any key, allowing BATCH to continue reading the input file.

BATCH then executes the DIRECT command, which lists the directory of RXA1 on the line printer. Since spooling is active because RXA0 was specified as the spooling device in the SUBMIT command line and a non-file-structured device is specified as output in the DIRECT command, BATCH intercepts this output and stores it in a temporary file on the spool device. The output is stored in a file named BTCHA1. BATCH then outputs the message

```
#SPOOL TO FILE BTCHA1
```

on both the console terminal and the line printer, if available. If another file is routed to the spool device, it will be assigned the file name BTCHA2, and any successive files will be named in the following sequence:

```
BTCHA3
```

```
.
```

```
.
```

```
.
```

```
BTCHA9
```

```
BTCHB0
```

```
.
```

```
.
```

```
BTCHZ9
```

A total of 260 spool files are permitted. If output to a spool file is generated by a program that appends a default extension to output file names, the spool file will be assigned a standard default extension. All of the spool files may then be transferred to the terminal or line printer by using the TYPE or LIST command with the input file specification dev:BTCH??.*.

You may type CTRL/C at any time during a batch processing run. Typing CTRL/C at the program level causes an effective jump to location 07600, which recalls the BATCH program. BATCH program then recognizes the CTRL/C and terminates the BATCH run. Sometimes two CTRL/C's are required to be typed in succession to return to the monitor.