

The application has been primarily designed to be operatorless in its execution. However, a manual override capability has been provided. No changes in existing application programs or JCL is required.

JOB PROCESSING FUNCTION

Job processing for batch jobs can be broken down into several tasks:

1. SCHEDULING. This consists of determining when to run each job that is on the Master Job list. The Master Job list is the set of all batch jobs whose processing needs to be scheduled (reasons for scheduling will be discussed). Along with the scheduling of jobs, interdependencies among jobs must be taken into consideration. Certain jobs' processing will be "dependent" upon the successful completion of other jobs. If the job stream that processes Labor vouchers aborts, we would not want to begin the processing of our Labor Variance job stream until we successfully finish the Labor Vouchers. Scheduling also includes identifying jobs that require "extra considerations" such as tape mounts or private volume mounts.

2. PROCESSING. This includes 2 interrelated tasks:

a. Streaming the jobs on the schedule. For the next job to be processed, a check must be made for the successful completion of all jobs this job is dependent on, check to make sure all extra considerations have been taken care of (private volumes, tapes, etc.), and then stream the job.

b. Monitoring the processing of each job. Processing must be monitored so that when a job finishes or aborts,

processing continues with either a restart of the aborted job or the streaming of the next available job.

3. REPORTING. After all processing is completed, the results must be recorded onto the master schedule and reported to the system users. In Data Processing it is necessary to inform the users as to what was accomplished and to inform the systems group of what problems must be fixed.

AUTOMATION OF THE JOB PROCESSING FUNCTION: ACE

We have found that almost all of the job processing function can be automated. ACE is our applications subsystem that handles ALL of our schedule processing for our Accounting department. ACE is an online job-scheduling and processing application. The application controls simultaneous processing queues of jobs in a multi-machine network. A single master schedule controls the queues and the machines on which execution will occur. ACE is operatorless in its execution and requires NO changes in existing application programs or JCL to implement. The subsystem has two online programs, SCHEDULER and PROCESSOR, that are used to create and execute the job schedules.

Taking the job processing function described above it will now be shown how ACE is set up and executed to process jobs. Figure 2.1 shows an overview of the implementation of the ACE subsystem.

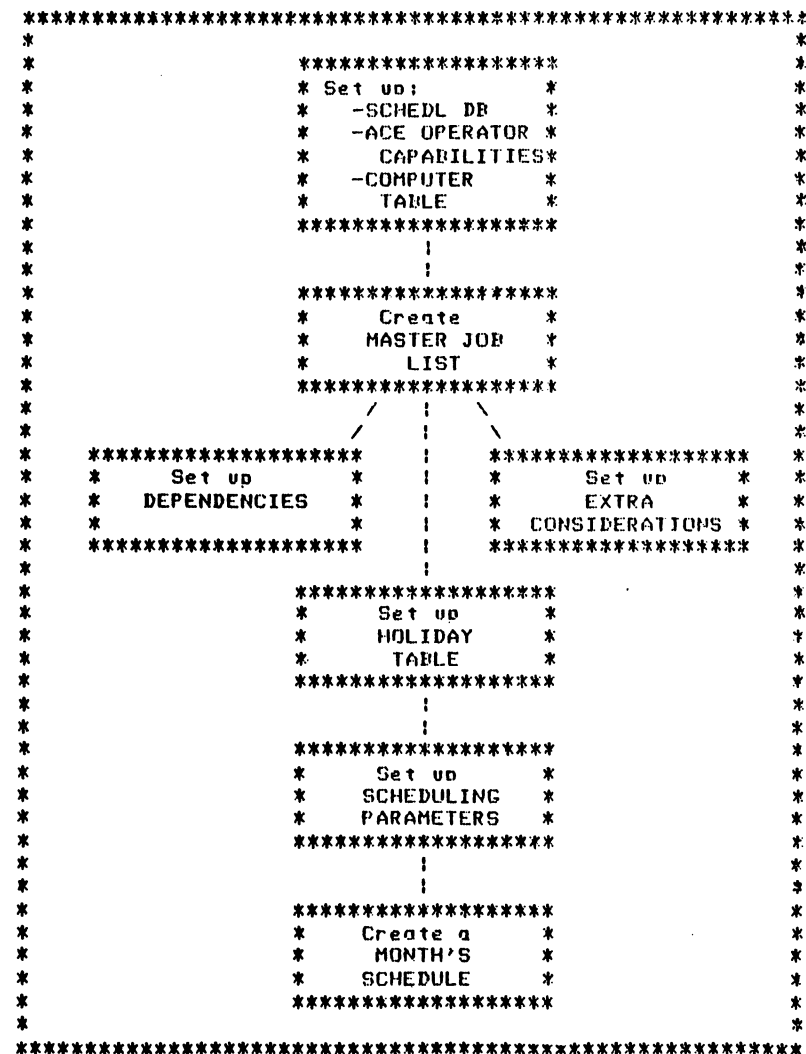


Figure 2.1 Implementing ACE

ACE: THE MASTER JOB LIST

ACE begins with the creation of a MASTER JOB LIST. This list contains all the jobs (on all the machines) that ACE will control. To create the list only two pieces of information are necessary, (1) the fully-qualified name of the file in which the JCL resides, which will be called the "JOB-NAME", and (2) the computer on which the JCL usually resides. Optional information includes "DEPENDENCIES", "EXTRA-CONSIDERATIONS", START- TIME-STOP-TIME WINDOW, DEFAULT-QUEUE, AND DEFAULT- SEQUENCE NUMBER.

Dependencies are the job-names of those jobs upon whose successful completion the current job depends. Extra considerations are dependencies that ACE can't handle. The application can check to see if the proper private volume is currently mounted, but it cannot physically mount the private volume itself. So the extra consideration is the reminder to the person setting up the daily schedule that "PRIVATE VOLUME AUDIT01 IS NEEDED FOR THIS JOB". Extra considerations are also commonly used to remind of needed tape mounts. The start-time-stop-time window allows a scheduled job to be run only between two specified times, i.e. the "window". If the window cannot be satisfied, the job is not streamed. The default queue and default sequence number allow a one-time specification of these items when their values are fairly static.

ACE: CREATING MONTHLY JOB SCHEDULES

Once the Master Job List has been specified the monthly job schedules can be created. To aid in the scheduling of jobs a set of "scheduling parameters" are defined within the system and can be used to create a schedule.

In our Accounting department we process Labor Vouchers every Wednesday night. We process our first General Ledger to microfiche on the 3rd working-day of every month. We process some jobs every other week on Monday, some jobs every day of the week and some jobs only twice a year. What we have found is that most of our jobs have a distinct pattern to their schedule. We've defined a set of parameters that describe these patterns to aid us in our monthly scheduling. These parameters allow for scheduling a job to be executed by the day of the week, the workday of the month or the particular month of the year. Using combinations of parameters any pattern can be described. There are also parameters that allow for the exclusion of days, weeks, or months.

Once the parameters have been specified for each job, creating a specified month's schedule consists of "applying" these parameters to the month's calendar. The result is the actual schedule of the date each job will run during that month. ACE maintains a table of holidays (inputted by the user) and will not schedule jobs on those specific dates. Once satisfied with the schedule for the month, and when appropriate (e.g. the beginning of the month), the schedule is activated for use.

ACE: DAILY JOB PROCESSING

Daily processing begins by entering SCHEDULER and requesting the day's schedule. All jobs scheduled for this day, and all jobs from previous schedules that either aborted or were "deferred" will be presented. Only when a job finishes successfully or is cancelled is it removed from the current schedule. Sequence numbers and queue numbers must be assigned to each job on the schedule. Dependencies are reviewed and are "deactivated" or additional added if desired. For jobs requiring private volumes or tapes, the private volume name or the logical device number on which the tape resides can be entered into the system. (PROCESSOR will then check to make sure the private volume is mounted before streaming a job and in the case of tapes will reply to the first tape request). Each job that appears on the schedule can either (1) be left scheduled to run this day, (2) deferred for future processing, or (3) cancelled from the schedule. Deferred jobs will continue to appear each following day until scheduled or cancelled. Cancelled jobs will not appear again until the next date they are scheduled to run. Jobs that were not scheduled may be added to the schedule at this time as long as they exist on the Master Job List. For all jobs that are left on the schedule, MPE passwords are entered and the schedule is ready to process.

The program PROCESSOR (Figure 3.1) executes the schedule. PROCESSOR is an online program that can be run from any terminal on the system. The program is executed in one of two modes:

- (1) AUTOMATIC RESUME mode in which PROCESSOR will execute the entire schedule without need of any user interaction. This is the pure "operatorless" mode.
- (2) MANUAL RESUME mode in which PROCESSOR will execute the entire schedule by stopping after each job is streamed and waiting for the user to let it stream its next job. This mode is extremely useful when many jobs are dependent on a single job. One can let the first job run and then monitor the job to completion before letting PROCESSOR pick the next job. Should the key job abort the user could fix the problem and restart the job so it finishes successfully. This way the jobs depending on its completion would be streamed instead of being deferred because their dependencies couldn't be met.

The program allows the user to switch between automatic resume and manual resume at any time by using control-Y. The user could monitor that first key job using manual resume and upon its successful completion leave PROCESSOR in automatic-resume and go home.

PROCESSOR picks the next job to stream according to a straightforward set of rules.

- (1) Only one job from a particular queue can be in processing at any time. Until the outcome of the current job is resolved, all following jobs in that queue wait.
- (2) Each queue is processed in the order of the sequence numbers assigned to each job. No job will be considered

(3) PROCESSOR will continually loop sequentially through the queues checking to see if the next job in each queue is ready to stream or must be deferred because of dependency aborts. PROCESSOR can have as many jobs processing as there are queues active for the schedule (it can't be more because of rule (1) and it could be less because dependencies can cross queues forcing one queue to wait until another's job finishes).

```

*****
* MAIN MENU *
*****

```

```

*****
*
*****
*
*****
*
*****
*
**RUN SCHEDULE**      **LIST SCHEDULE**      **CHANGE SCHEDULE**    **RUN SYS UTIL**
*****                *****                *****                *****
**AUTOMATIC          **LIST SCHEDULE          **SUSPEND A QUEUE       **STREAM A JOB
RESUME               ONLINE/OFFLINE           *DELETE A QUEUE        *RUN TOP
**MANUAL              *LIST A QUEUE            *ADD, DELETE, OR       *RUN SPOOK
RESUME               ONLINE/OFFLINE           MODIFY A JOB           *EXECUTE AN
                                  MPE COMMAND
**LIST A JOB
ONLINE/OFFLINE
*'SHOW' COMMANDS
   SHOWJOB JOB=@J
   SHOWJOB
   SHOWOUT JOB=@
   SHOWOUT JOB=@J
   SHOWOUT JOB=@:READY.N

```

Besides being able to specify which mode to run the schedule in, there are other tasks which can be performed from PROCESSOR.

Second, the schedule can be altered from within PROCESSOR. Queues can be deferred, cancelled, or suspended. Jobs can be added, modified or deleted. When modifying jobs, dependencies can be added or deactivated.

Third, PROCESSOR uses process handling to allow the user to stream a job, run TDP,PUB,SYS, run SPOOK,PUB,SYS, or execute other APE commands. This is to allow for restarting aborted jobs from within PROCESSOR. SPOOK is used to read spoolfiles, TDP is an editor, and the stream command not only streams the job but also sets up the job streamed as an ACE job.

QUEUE	PRIORITY*	STATUS	JOB-NAME	DEPENDENCIES/ EXTRA CONSIDERATIONS
01*10*01	R		JOB11.GROUP.ACCOUNT	**JOBXX.GROUP.ACCOUNT **JOBY.Y.GROUP.ACCOUNT JOBZZ.GROUP.ACCOUNT
01*20*02	S		JOB12.GROUP.ACCOUNT	**JOB11.GROUP.ACCOUNT NEEDS TAPE MOUNT FOR LAST STEP

Figure 3.2 Current Schedule List

JOB11.GROUP.ACCOUNT	STATUS-CODE:	SCHEDULE-DATE:
QUEUE-NUMBER: __	DEFAULT-QUEUE-NUMBER: __	
PRIORITY: __	DEFAULT-PRIORITY: __	
COMPUTER-NUMBER: __	DEFAULT-COMPUTER-NUMBER: __	
START-TIME: _____	START-TIME-WINDOW: _____	
STOP-TIME: _____	STOP-TIME-WINDOW: _____	
AVG-RUN-TIME: _____		
	1 2 3	
	12345678901234567890123456789012	

THIS MONTH'S SCHEDULE:	0000R0AR0000DR010000010000010001	
DEPENDENCIES:	JOBXX.GROUP.ACCOUNT	
	JOBY.Y.GROUP.ACCOUNT	
	JOBZZ.GROUP.ACCOUNT	
EXTRA CONSIDERATIONS:		

JOB11.GROUP.ACCOUNT INFORMATION:

Figure 3.3 Current Schedule Job Information.

When PROCESSOR finishes processing the production schedule the last job streamed generates a report summarizing the results of the processing. Aborted jobs, deferred jobs, cancelled jobs, and successful jobs are all reported in this report. We then post this report for the users.