



Drive Your Enterprise with Event-Driven Scheduling

How much time does your system operator spend creating and managing timed jobs for your enterprise's nightly operations? How is your job schedule affected if a job runs late or doesn't run at all? Can your company afford to have an extra system operator on staff just to make sure nightly processing completes accurately and on time?

Robot/SCHEDULE Enterprise® addresses these problems with Event Monitors. Use Event Monitors to create an event-driven schedule for your enterprise using system events on your Windows, UNIX, and Linux servers to launch reactive jobs through Robot/SCHEDULE®.

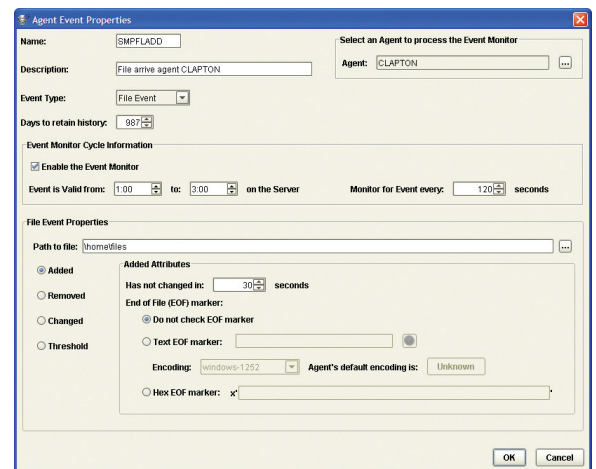
Robot/SCHEDULE Enterprise is completely integrated with Robot/SCHEDULE, so you can use the Robot/SCHEDULE Explorer to set up and manage your Event Monitors. The same easy-to-use tools you're already familiar with—job blueprints, schedule forecasts, the Schedule Activity Monitor—make it simple to monitor your enterprise jobs and Event Monitors. Robot/SCHEDULE maintains a complete history of monitored events, saving time if you need to diagnose a processing problem quickly.

The scenario below primarily describes how you can use files arriving on a server as a prerequisite to launch other jobs across your enterprise. You also can use Robot/SCHEDULE Enterprise to monitor changes to a file or directory, a file that's not growing, an application or a daemon starting, or your own user-defined events on your servers.

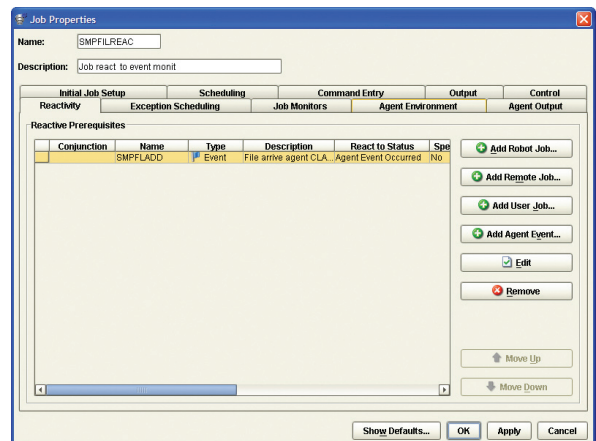
THE SCENARIO

The following scenario depicts how a retail store's nightly job schedule is dependent on manual and timed jobs, and the problems that can occur using this method.

It's 9:00 p.m. in the local mall and the long retail day is over. But, the behind-the-scenes operations are just beginning. Each day, your company's data center operations group must gather the day's transactions, consolidate data from 125 stores,



Creating an event-driven schedule is easy with the Robot/SCHEDULE Enterprise. In this example, the event monitor is monitoring a directory for a file to arrive on the Agent system named CLAPTON. Below, a Robot/SCHEDULE job uses the event monitor as a prerequisite to run.





upload the data to the retail software running on the IBM® System i® (AS/400®), run batch processes, back up the data, generate reports, and send a new inventory file back to each of the 125 stores. No problem, right?

Now, take a closer look at this scenario and see if you notice any problems. Currently, the process of sending the transaction file starts when the local store manager closes the doors and pushes a button on the point-of-sale (POS) terminals. At the corporate servers, a process starts at 10:30 to allow time to collect the data from every store. Five Windows servers receive data for up to 25 stores each. The Windows servers use the Microsoft scheduler to send the data to an AIX server. The AIX server runs a job via Cron to consolidate the data before a scheduled FTP process sends it to the System i.

At midnight, the System i starts its batch process to get the consolidated FTP file. The IBM Scheduler runs a job at 12:15 that starts posting the day's transactions to the retail inventory and sales systems. By 2:00 a.m., backups start on the System i, which usually last an hour. At 3:00, the final reporting job runs and the new inventory file is sent back to the five Windows servers via FTP. The Windows servers have a scheduled job that runs at 4:00 to send each of the 25 stores a new transaction file.

THE PROBLEM

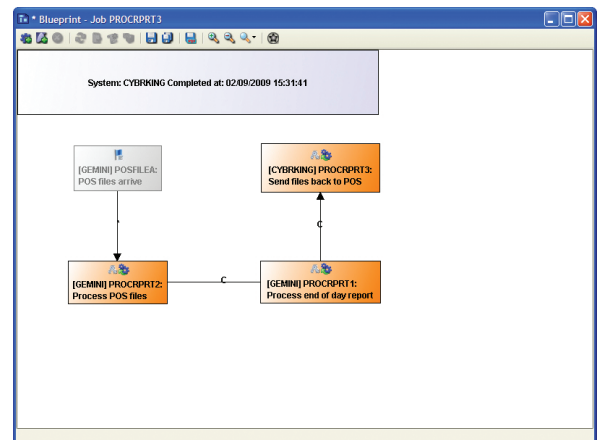
Did you see any problems in this process? Everything works great if all the timed events happen as planned; but what if one store is late sending the initial file? What if one of the Windows servers locks up? What if the Cron job runs before all of the Windows servers have sent their transaction files? What if the operator doesn't notice a "mount next tape" message on the System i?

Will an operator notice any of these problems in time? Can an operator restart any of the processes? Is all the activity logged manually every night? How long before a PCI or SOX auditor reviews these processes? What happens if the company wants to add more stores—does this process have the capacity?

THE SOLUTION

Robot/SCHEDULE Enterprise, the enterprise job scheduling software, solves all of these issues—and more—by eliminating timed events. Using Robot/SCHEDULE Enterprise in our scenario, each Windows server has an event monitor that notices, within seconds, that the file transfer from one of the 25 stores has arrived. Once all 25 prerequisite events (file transfers) have occurred, a reactive job in Robot/SCHEDULE sends the data to the AIX server. An independent process on each of the Windows servers runs to verify that this process completes by a certain time. The on-call staff is notified automatically if a process is running late.

On the AIX server, five separate file event monitors check for Windows file transfers. Once all five files arrive, the reactive job processes and consolidates the data.



The job schedule blueprint shows an event-driven schedule that starts when the POS file arrives on an agent system.



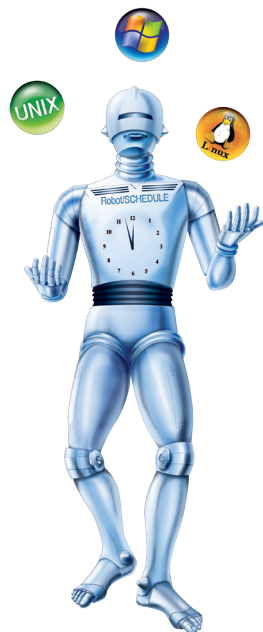
On the System i, a reactive job starts to post the transactions to the sales and inventory systems as soon as the AIX job is done with consolidation. The sales and inventory jobs are prerequisites for the backup job. When the backup job completes, another reactive job submits the report jobs. These report jobs are the prerequisites for sending the inventory files back to each of the Windows servers.

Once each Windows server has its file, it launches 25 processes to send the new transaction files to the Linux POS terminals.

At the end of the process, the administrator receives a report from Robot/SCHEDULE that shows the completion status and times for the evening.

CONCLUSION

The retail business in this scenario can grow easily because Robot/SCHEDULE Enterprise has eliminated about three hours from the night processing by using an event-driven schedule, rather than timed and manual jobs. Event-driven scheduling takes away the guessing game for enterprise scheduling. And, Robot/SCHEDULE Enterprise ensures that your jobs run when they're supposed to—even when the triggering event is on another server.



Robot/SCHEDULE Enterprise brings the power of Robot/SCHEDULE to your Windows, UNIX, and Linux servers. Learn how to create event-driven schedules based on system events across your enterprise.

help/systems

952/933-0609

www.helpsystems.com