

# JOB SCHEDULING VS WORKLOAD AUTOMATION: WHAT'S THE DIFFERENCE?



Most

IT managers understand the differences between job scheduling and enterprise workload automation software, so there's no point in explaining it further, right?

The problem is that even though you may know the differences, others may not. And at some point, you'll probably have to explain those differences to someone else, such as a programmer, a Service Desk tech, your CIO, controller, or CEO.

For those times, here's a simple explanation of the differences in plain English.

## Job Scheduling Software

Job scheduling packages are used for controlling the unattended processing of batch jobs on a single system. All major OS platforms have a job scheduler built into their operating system, such as the Windows Task Scheduler, the integrated IBM i job scheduler, and the cron job scheduler for Unix-like environments and Linux. For the IBM i operating system, there are also several third-party job scheduling packages that provide capabilities beyond the basic IBM i job scheduler (see my [community post on IBM i job scheduling software](#) for a more complete list of vendors offering third-party schedulers).

Job schedulers are generally platform-specific. They are configured to submit batch jobs for execution according to a pre-defined schedule or after a dependent event occurs. Some examples of dependent events that can submit jobs in a job scheduler are specific days and times of the week (i.e. Monday at 10:00 AM or Tuesday at 3:00 AM), specific days of the month (i.e. last date of month; first Monday; the 15th), or the completion of another batch job.

Traditional job scheduling software generally runs jobs only on one machine, which introduces a number of problems for organizations, including:

- **Siloed solutions.** Lack of coordination between job schedulers running on different operating systems, such that critical related jobs on different operating systems may not run or may run out of sequence. Job schedulers generally don't talk or work with each other across platforms.
- **Complexity.** Scheduling jobs and maintenance becomes more complicated as several schedulers on different operating systems must be manually configured and maintained, increasing complexity and the possibility of errors. More time from IT personnel is needed to perform these duties.
- **Manual intervention.** Job schedulers frequently require manual intervention to correct problems between related scheduled processes on different machines, such as when a file is created on one machine and FTP'd to a second machine to be processed.
- **Programming resources.** Job scheduling frequently requires additional scripting or programming to fill in gaps that occur when coordinating processes between machines and operating systems.

## Enterprise Workload Automation Software

Enterprise workload automation software offers improvements to these challenges. It performs all the same functions as job scheduling software, with these critical enhancements for working with batch jobs and coordinating processes on different servers and operating systems:

- **Consistency.** One standard graphical interface for configuring workloads on different servers, rather than having to configure job entries in several different packages.

- **Visibility and control.** Centralized control for scheduling and running different workloads on different servers, providing an enterprise-wide view and control over all batch jobs across the enterprise. Enterprise workload automation software also allows you to view active jobs and job run histories for processes controlled by the software.
- **Simplified file transfers.** Scheduling and integration of file transfers between different platforms and machines from the central interface, rather than scheduling transfers on each individual server.
- **Faster workload deployments.** Automatically promote new workloads from test to QA to production, eliminating the need to repeat manual configurations when going live (which can cause manual configuration errors).
- **Big data.** Schedule big data workloads using [Hadoop](#).
- **Programming reduction.** Elimination of the need for manual scripting or programming when using different job schedulers on different systems. Job dependencies are worked out in your enterprise workload automation software, not in your programming department.
- **Error reduction.** Improve coordination between scheduled processes for co-dependent processing running on different servers, reducing errors that can occur when jobs run out of sequence or a critical job fails.
- **SLA improvement.** Integrate schedule changes to avoid resource conflicts and forecasted job completion times, ensuring that jobs will finish on time, helping to satisfy [Service Level Agreements](#) (SLAs).
- **Reduced staff time needed to run scheduled jobs and processes.** Less time is needed to configure, monitor, and respond with a single-level workload automation system than is needed for multiple separate job scheduler installs.

Enterprise workload automation solutions are to job scheduling software what a 2017 Tesla with AutoPilot is to a 1966 Volkswagon Beetle. Both products do the same basic job (scheduling and running of jobs and processes), but you can do more, and do it much better, with an enterprise workload automation solution. An enterprise workload automation solution is much more than a souped-up job scheduler: it's a significant improvement in the entire implementation of job and workload scheduling.

## To learn more

This post provides a basic understanding of what job schedulers and enterprise workload automation solutions do and what the advantages are in using an enterprise workload automation solution. To find out more about enterprise workload automation products such as [Control-M solution](#), please contact [BMC Software](#) for additional information.