

Ten Secrets of Job Scheduling

Selecting the Right Job Scheduling Vendor

Executive Summary

As the world turns more and more to graphics devices, production job scheduling seems to be less and less important.

The reality is precisely the opposite – production job scheduling is actually becoming more and more important.

Here are just a few of the reasons:

- International workloads – jobs can be scheduled from different countries around the clock
- Demand for true 24 by 7 by 365 operations (Mon to Fri, 9 to 5, is ancient history)
- Greater back office loads and the concomitant need to automate
- Distributed NOC and control centers
- **AWS is revolutionizing computing and automation is critical for successful AWS implementations**

This brief white paper lists the ten most important secrets to consider when selecting your production job scheduling vendor.

Secret 1: Be AWS Enabled – Really Enabled

Running production jobs on AWS is far, far more than simply defining an account and installing a product.

What is essential is a **single pane of glass** where all legacy jobs and all AWS jobs can be seen on a single 60-inch monitor.

Central control for hundreds of queue engines, both those running on local hardware and those queue engines running in the cloud.

And don't forget scaling – too many vendors loudly tout their AWS expertise, but upon investigation it turns out their implementations do not scale.

Ask the vendors being evaluated to **show you a run rate of one million (10^6) jobs per day in AWS** – you can quickly separate the wheat from the chaff.

Secret 2: Easy Scalability

What does “scalability” mean? Ability to grow.

From 500 jobs per day to two million jobs per day (that is, 23 jobs PER SECOND).

As workloads rapidly grow, it is essential that a virtual machine can be spun up and put online in minutes. Adding a queue engine to the production job scheduler should be a trivial and simple step and involve no interruption to the current production environment.

This is even more important in your rapidly growing AWS implementation.

When the queue engine is added to AWS or a new local machine, the central scheduler should automatically recognize the new machine.

Secret 3: Support of Distributable Execution Across Multiple Platforms

In a perfect world, all machines are the same make and model with the same operating system.

In the real world, a job scheduling environment is typically a potpourri of widely varying platforms with wildly different performance and even more diverse connection protocols.

It is essential the selected job scheduling vendor can support the common platforms of Windows and Linux, but needs to also support the still widely used legacy platforms of Unix, Solaris, HP-UX, SCO, and AIX.

A special note should be taken of support of iSeries machines as, even today, these carry a surprisingly large job scheduling workload.

Secret 4: Single Pane of Glass Management and Monitoring

With a wide array of disparate servers and operating systems dotted across the globe, the central job scheduling operator console becomes even more crucial.

It is essential that the job scheduler product provides the operators with both macro and micro views – from the current state of the entire worldwide job mix on 100 servers down to an individual job - in just a few mouse clicks.

With thousands – or tens of thousands -- of jobs running in a day, effective and well-designed filters are essential to allowing the operators to focus on the jobs being reviewed.

In today's 24-hour-a-day world, secure access remotely from mobile devices is essential.

Secret 5: Robustness Through High Availability

No single point of failure can exist in the selected job scheduler.

High Availability is no longer a luxury – when the central scheduling server fails or loses its network connection, it is essential that a cascade of secondary scheduling servers seamlessly take over the workload, automatically and immediately; the days of laboriously manually swapping the central job scheduling server are long gone.

Secret 6: Intelligent Enterprise-Level Load Balancing

When deploying dozens – or even hundreds – of servers around the world, load balancing becomes indispensable.

Without intelligent load balancing at the enterprise level, some servers are left idle while others are running at full capacity. This in turn leads to jobs starting late and ending late and extremely irate end users – the internal customers of the job scheduling product.

Secret 7: Comprehensive Alerts and Notifications

The production job scheduler selected must ensure that all issues that arise during the execution of a job are immediately alerted.

Failing to execute a job, jobs ending with error codes, jobs that are late to start, and jobs that are late in completing are some of the common issues busy operators need to be alerted about.

Successful completion of a critical job should also be able to trigger an alert.

The product must also employ alerts for environment problems like a database server going offline or an execution engine becoming unavailable.

Alerts should be issued via email, alphanumeric pager, SMS, pop-ups, beeper, and Twitter.

Secret 8: Total Security Integration Through Microsoft Active Directory

With good reason, Microsoft's Active Directory has become the *de facto* standard for authentication, and not just for jobs run on Windows servers. It is crucial that a job can be configured to run using specified Active Directory credentials. It is also useful for the job scheduling vendor to provide alternative means of specifying the credentials under which the job runs. It is also extremely important that the product provides the ability to set an Active Directory account at the level of job class. (Some production job scheduling customers have thousands of job templates.)

Along with job credentials, the vendor needs also to provide granularity of access and viewing rights – obviously operators need full access, but the end user should be limited to just Read Access.

Secret 9: Powerful, Easy Calendars

The heart and soul of any production job scheduler is the power and the ease-of-use of the calendar.

Apart from the regular scheduling cycles, such as daily, weekly, monthly and yearly, the selected product needs an advanced calendar for periods, holidays, and recurring days like second Wednesday of every month. It's essential that any condition can be met; for example, the Last Work Day of the Month should be a trivial matter of clicking one check box.

Secret 10: Powerful, Easy Job Dependencies

Along with calendaring, job dependences are high on the list of essentials for a production job scheduler. A job can be configured to have any number of Job, ODBC or File dependencies. A job can be configured to run according to the return code of a parent job, or when a database value has been modified, or when a file has been copied or modified.

For example, a common situation is when files are dispatched from remote warehouses or stores; only when all the files have been delivered can the nightly jobs start. So far, so good. But there need to be facilities to start the nightly run even if all the files have not been delivered by the cutoff time; all people experienced in real world job scheduling take this as a given. (See next point.)

And One More Thing... Seasoned, Local Support

Far too often U.S. and European companies outsource their support to Third World countries to save money. And they do save a lot of money because world-class local American and European support people are expensive.

But these vendors do not realize the extremely critical nature of job scheduling, as these jobs run the business, be it payroll, inventory, or the nightly consolidation from remote warehouses or stores, etc.

As you move to AWS, ensure the vendor has local people who are well versed in AWS and especially job scheduling on AWS.

In the evaluation and the proof of concept stages, insist on a local engineer to be on site – at no cost.

Please contact Argent for your free job scheduling checklist

Argent Software created this document for informational purposes only. Argent Software makes no warranties, express or implied, in this document. The information in this document is subject to change without notice. Argent Software shall not be liable for any technical or editorial errors, or omissions contained in this document, nor for incidental, indirect or consequential damages resulting from the furnishing, performance, or use of the material contained in this document, or the document itself. All views expressed are the opinions of Argent Software. All trademarks and registered trademarks are the property of their respective owners.