

A Clean Slate for Enterprise Scheduling

By Pat Cameron

In the past, most enterprise scheduling or workload automation products were created using a client-server strategy. This “older” technology consisted of software on a host server (often, a dedicated server), software on the desktop, software on individual servers, and a database. This software now represents the new era of legacy applications—technology that was often so difficult for organizations to deploy and maintain, that it failed to provide a reasonable Return on Investment (ROI).

Disadvantages of Client-Server Scheduling Tools

Besides cost, there are several other disadvantages to the client-server approach:

♦ Too Much Software To Maintain

A big logistical issue is the need to install software on every server and workstation involved. Usually client-server technology is not platform-independent, so you must make changes on each platform where you install the software. Imagine having to continually install patches on multiple servers. Before you know it, you need a group of people dedicated to maintaining the scheduling tool.

♦ User Interface Blues

A second issue is the user interface. Most interfaces could run only on the Windows operating system, which meant installing software on every workstation that needed access to the scheduling software. Often, each workstation also needed run-time software for Java, or similar technology, which can be a big hurdle for large implementations.

♦ Lack Of Security

The third big concern is security. Many client/server enterprise schedulers rely on the security of the host server to establish roles for the software. As a result, the support team had to become security experts for each platform. In today’s world, security is not optional: enterprise-scheduling products must offer more than just operating system-level security. The processes that run across your enterprise are critical and it can be dangerous if your schedule is open to access.

♦ Company Acquisition Software

A fourth issue that affected both quality and reliability resulted from the increase in acquisitions. Many of the main players in the market would acquire a product and do little except to rename it. Customers think they are getting an integrated, cross-platform solution. In reality, they are simply getting multiple interfaces to use to build their schedule. Acquisitions are not bad—but it’s difficult to merge applications. And, acquisitions are usually profitable only if the acquiring company can keep the maintenance streams flowing. To do that, they must keep current customers happy. Once a customer learns their favorite software isn’t being maintained, they quickly become unhappy.

♦ Patching Old Technology

The legacy issue also causes development and maintenance pain for organizations trying to meet audit requirements and consolidate their schedule. Developers like to rewrite parts of an application, so you often end up with a patchwork of features appended to the base product. Many of these features would be better, or different, if the development team could have started over. But they can't start over because it's cost-prohibitive. And, customers are often unhappy if a developer removes a feature or changes it significantly.

♦ No Built-in Notification

There are other issues, of course. Service Level Agreements and business management pressures require easy, cost-effective notification options. Many client/server enterprise-scheduling options don't have built-in notification. So, users must buy or develop their own paging, e-mail, or SNMP-compatible software. It's not easy for legacy applications to take advantage of free-class objects (as in Java) and to integrate standard industry notifications.

♦ Too Many Scripts

It's common to price enterprise-scheduling solutions based on the number of jobs the scheduler controls. This is a bad idea for automated scheduling. We've always advocated the idea of a granular scheduling. In fact, we tell customers that having more jobs is often better. For one thing, if you charge by the number of jobs, customers create custom scripts that try to do too much. They end up with complicated scripts that can be difficult to run. And, some client/server applications allow only one script per job, which can force the customer to write extra code.

Now, there is a better solution.

Skybot Scheduler

Lately, we hear more and more about Cloud computing and "Software as a Service" as the new software model. The software applications built for this model use a user interface that supports Web browsers across multiple operating systems—Windows, Linux, UNIX, and Mac OS—and takes advantage of the latest Web technologies.

The first benefit of this model is that you don't have to install software for each end user. The host server can run the application and serve the user's browser interface through HTTP. Generally, these solutions use new web technology to deploy the application. You can develop enterprise scheduling and workload automation software this way, as well as payroll, ERP, e-mail, and accounting software. To control enterprise-wide processes with a scheduling application, you usually need to have agent software running on each server.

So, when a company with more than 25 years of enterprise-scheduling experience decides to build a scheduling application using this technology, the marketplace experiences a very robust product: Skybot Scheduler.

Web Browser Interface

Skybot Scheduler overcomes many client/server issues by supporting multiple Web browsers, including Internet Explorer and Firefox. An end user can administer Skybot Scheduler from anywhere with just

browser access and a login to the server that hosts the Web application. Both the server and the agent software were developed in Java, and Skybot Scheduler has its own database, which makes life easier for everyone.

Improved Security

Skybot Scheduler improves network security, because all of its “conversations” are encrypted. It also offers its own security that supports Operator, User, Administrator, and user-defined roles, and can be based on different objects in the product. You can segregate lines of business, or customers, and host all your jobs on one server. Users see only the jobs relevant to their role.

Easy to Set Up and Maintain

The only thing your agent servers need is a Java Virtual Machine (JVM). You just set up the host server, install the JVM and the agent software that communicates to the host server, and start your Web browser. You’re ready to go with a solution that’s easy to set up, easy to use, and easy to maintain.

Event-Driven Scheduling

Skybot Scheduler supports event-driven scheduling across your network based on system events such as file or directory changes, daemons, services, SNMP traps, manual events, and tasks starting, finishing, or failing. It offers advanced scheduling features for end-of-day, end-of-week, or end-of-month processing, plus event monitoring. It even lets you run jobs at specific intervals, a fixed number of minutes, days, or weeks.

Audit-Ready Capabilities

These days, it’s easy to be overwhelmed with regulations. Don’t worry—Skybot Scheduler has built-in auditing, job and server history, separation of duties, and more to help you pass IT audits easily. And, you can easily export historical data into your favorite spreadsheet or business intelligence solution.

Built for Today’s Marketplace

With more than 25 years of scheduling experience, it’s safe to say that we know the needs and wants of the job-scheduling marketplace. Skybot Scheduler is a secure job scheduler that’s easy to set up, easy to use, and easy to maintain. It spans your enterprise of Windows, UNIX, and Linux servers with a fast, portable Web browser interface, and the auditing capabilities needed for today’s regulations. Skybot Scheduler is definitely a fresh start for enterprise schedulers.

Visit www.skybotsoftware.com or call
1.877.506.4786 for more information
or a FREE 30-day trial.