

Realtime  
publishers

"Leading the Conversation"

# *The Shortcut Guide<sup>™</sup> To*



# IT Workload Automation and Job Scheduling

*sponsored by*



*Mark Scott*

Chapter 2: IT Workload Automation and Job Scheduling.....17

Defining Job Requirements.....18

    Focusing on the Outcome of the Automation Jobs.....19

    Working with the Resources.....20

    Prioritizing the Workload .....21

    Working with Job Interfaces.....21

        Dealing with Security .....22

    Monitoring the Workloads.....23

    Overseeing Plan Orchestrations.....24

Time-Based Scheduling.....24

    Working Within the Time Constraints.....25

    Working with the Business Calendar.....26

    Adapting to Resource Constraints .....27

    Running Tasks Synchronously vs. Asynchronously.....28

Event Driven Scheduling .....29

    Working with On-Demand Interfaces.....30

    Dynamic Utilization of Resources .....31

    Monitoring On-Demand Job Tasks.....31

Job Execution.....32

    Working with Server Topologies.....33

    Changing with the Environment .....34

    Load Balancing .....35

    Error Correction and Compensation .....35

Monitoring IT Automation Workloads.....36

    Alerting Functions .....36

    Reporting.....37

    Auditing .....37

Summary .....38

Download Additional eBooks from Realtime Nexus!.....38

## Copyright Statement

© 2008 Realtime Publishers, Inc. All rights reserved. This site contains materials that have been created, developed, or commissioned by, and published with the permission of, Realtime Publishers, Inc. (the “Materials”) and this site and any such Materials are protected by international copyright and trademark laws.

THE MATERIALS ARE PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. The Materials are subject to change without notice and do not represent a commitment on the part of Realtime Publishers, Inc or its web site sponsors. In no event shall Realtime Publishers, Inc. or its web site sponsors be held liable for technical or editorial errors or omissions contained in the Materials, including without limitation, for any direct, indirect, incidental, special, exemplary or consequential damages whatsoever resulting from the use of any information contained in the Materials.

The Materials (including but not limited to the text, images, audio, and/or video) may not be copied, reproduced, republished, uploaded, posted, transmitted, or distributed in any way, in whole or in part, except that one copy may be downloaded for your personal, non-commercial use on a single computer. In connection with such use, you may not modify or obscure any copyright or other proprietary notice.

The Materials may contain trademarks, services marks and logos that are the property of third parties. You are not permitted to use these trademarks, services marks or logos without prior written consent of such third parties.

Realtime Publishers and the Realtime Publishers logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners.

If you have any questions about these terms, or if you would like information about licensing materials from Realtime Publishers, please contact us via e-mail at [info@realtimepublishers.com](mailto:info@realtimepublishers.com).

[**Editor's Note:** This eBook was downloaded from Realtime Nexus—The Digital Library. All leading technology guides from Realtime Publishers can be found at <http://nexus.realtimepublishers.com>.]

## Chapter 2: IT Workload Automation and Job Scheduling

There are many automation workloads that IT must see executed regularly on the servers throughout their organization. As mentioned in the previous chapter, the task of coordinating a series of individual jobs across servers, platforms, applications, firewalls, time zones, and wide area network (WAN) links can be daunting. When the jobs depend on one another, IT needs a system that allows these different tasks to work together as if designed as part of a single program.

This chapter will explore the process of scheduling those tasks and orchestrating them to work together. There are several aspects to consider when choreographing the jobs throughout the enterprise:

- **Defining Plan Requirements:** Plan requirements should be outcome based. The end result is paramount, and the steps required only facilitate that end. When considering plan requirements, one must identify the input sources and tasks required to provide those inputs. The dependencies of the jobs on one another must be determined. The interfaces that jobs use to communicate with one another must be defined. One must choose the appropriate paradigm for linking jobs (time schedules, events, and so on). The jobs compete for the same computing resources, so mechanisms must be devised for determining priorities. A system of monitoring and auditing must be established.
- **Time-Based Schedules:** The oldest and most familiar paradigm for scheduling, basing things on a time schedule, can be anything but simple. The windows and time constraints must be taken into consideration. Implementing the business calendar—including weekends, holidays, month end closings and fiscal events—must be factored into the schedule. Determining which events are synchronous and which are asynchronous, which events can occur in parallel and which in series, and how to keep everything interoperating correctly becomes the key to building a dependable, workable schedule.
- **Event-Driven Scheduling:** Plans can be created more simply if one job can call another job directly through a programmatic interface. To develop event-driven schedules, the manner in which jobs are triggered and monitored must be defined. The jobs are not scheduled, so dynamic selection of available resources becomes most important. The system must employ event-driven interfaces, such as a Web service, message queues, file triggers, server systems alerts, and other targets that listen for specific job occurrences. The system must be designed to audit and alert to irregularities and the absence of expected events.

- **Job Execution:** The execution of plans associated with a schedule must be responsive to the needs of the organization. The load must be balanced across resources, either manually or automatically. Resource utilization must be monitored on a regular basis to help keep the optimal usage of resources. Error-handling should be built-in to the system and compensating resources employed to keep the schedule on track when failures occur. There must be the means for prioritizing a related stream of jobs that occur on different systems throughout the organization and reorganizing those jobs as they relate to one another. Service Level Agreements (SLAs) help form the basis of these priorities and are critical to keeping the system well ordered.
- **Monitoring and Reporting:** To govern enterprise solutions, monitoring becomes vital. Establishing guidelines and mechanisms for reporting on the execution and health of the entire automation system helps Operations validate that everything is running as designed. Alerts will help the staff respond quickly to errors and correct them within the established limits of the SLAs. The reporting can be used to satisfy regulatory and corporate compliance standards. And proactive analysis of the operational data can help the system adjust to impending risks before those risks become issues and create disruptions in the automation workflows.

## Defining Job Requirements

The first step in planning any system is defining the requirements. IT workload automation can be challenging because of the disparity of the systems that work together to complete a single plan. There is a tendency to see each individual job as a plan within itself: One system to produce a data export. A second system does a complimentary export. A third job moves the files to a staging area. A different system picks up the files and imports them into a database. Another job processes the data into a data warehouse. Another job is built to create a report. Although all these jobs represent a single plan that eventually will help the VP of Operations plan the schedule for the next 2 weeks, it feels like a series of disconnected jobs.



For the purposes of this discussion, a *job* is a single operation on a server, such as a backup, data export, or reconciliation process. A *plan* is a series of one or more jobs that complete a business goal, such as loading the data warehouse or preparing month-end financial reports.

If the IT automation system can be viewed holistically, the plan appears more like a single operation rather than individually disjointed jobs. This view changes the way the jobs are monitored and the ways in which Operations will oversee scheduling and execution. It affects the way the automation load is troubleshooted. In addition, this view can help Operations visualize the entire organization and open new ways to use the resources of the enterprise to complete the tasks.

To do so, an automation system needs to abstract the individual tasks as functions or routines in a program. With that technology in place, the automation planners can build a specification for processing the plan, and schedule the tasks into the workload for the enterprise servers.

### ***Focusing on the Outcome of the Automation Jobs***

Someone once said programming is defining the output, identifying the input, and then processing the input into the output. Often in IT automation, however, the only real consideration is the expected output for the immediate job. The outcome of a desired plan will help to determine the steps in the schedule and the time windows in which they need to occur.

When operating an automation system, it is easy to concentrate on the execution of the individual job tasks. The more automated the better. But keeping the individual jobs running on individual servers is not the ultimate goal of the plan. It is important to remember that the automation jobs all have design goals to accomplish important business goals.

When changes in the business occur, the schedule will change. From an Operations standpoint, this is often disruptive. The staff can become resentful of the changes because they need to rebalance complex schedules and make changes that must be tested and troubleshot. A well-tuned schedule can be difficult to balance in the first place; changing an already full schedule can be even more challenging. The problems can make Operations very reluctant to accommodate change (which frustrates the business users) or cause plans to run unreliably.

When plans do not run correctly, Operations may also become relaxed about correcting the issues. If the department is overworked and understaffed, job failures can add more burden to the department than they are prepared to bear.

The first step is to help everyone remain focused on the fact that the end result of the plan is really the top priority, not the execution of the job schedule or a single, isolated server task. If the Operations staff remains focused on the goal of attaining the final state of the plan, they can keep a more positive attitude toward their daily tasks.

The system used to execute the jobs and troubleshoot issues can make a great difference in helping the IT staff remain focused and motivated. A system that simplifies the troubleshooting of jobs can help the staff find the errors and correct them quickly. This will get the plan to run more quickly and reduce the burden on IT.

Systems that can adapt more easily to changes in the business requirements will also help the IT staff. A centralized system that allows them to make changes to plans with a consistent interface will help them make the adjustments quickly. If the system helps them automate, audit, and maintain the configuration changes, the system will thus reduce the paperwork and other record keeping that attends such changes. A strong, centralized monitoring system will also help Operations ensure that the changes have been executed successfully and that the desired outcomes are being achieved. Providing the Operations staff with tools that help them do their jobs effectively will improve morale and provide more consistent results in executing the IT automation workloads.

### **Working with the Resources**

There is a lot of work to be accomplished, and only so much time and so many resources available to accomplish it. The issue faced by Operations and Automation Planners is making the entire IT automation workload succeed within these constraints.

The first task is scheduling. The challenges of creating a successful schedule will be discussed later in this chapter. With a successful schedule in place, there needs to be a set of criteria that help Operations determine whether they are successfully executing the schedule. Just knowing that jobs have been completed may not be enough information to know that the schedule is working or to avert disaster when conditions change. Operations staff needs clear requirements that show what is expected of the system and measures that indicate when they are headed for trouble. A system that can report clearly on the parameters that the requirements name helps Operations ensure that the outcome of the requirements is achieved.

Most schedules work when they are first created (after the typical period of troubleshooting and working out the unexpected bugs). Over time, however, conditions change. A job may run in just 10 or 15 minutes when it is first created. Twelve months later, it takes the same job more than an hour to run.

This change is incurred for many reasons. The amount of data handled by the job may grow over time. The concurrent workload on the server diverts resources to other tasks than the scheduled one (for example, indexing file system content, de-fragmenting hard drives, replicating data with other servers, and so on). Network bandwidth is reduced or consumed by other operations. Users begin the server at times that overlap the processing window. All these types of events can make executing the schedule difficult.

The key to remaining on top of this is good monitoring. An automation workload system that provides a clear picture of what is executing and how it is executing will provide Operations and the automation planners with the information they need to make adjustments and keep the schedule executing as required.

 There is more information about monitoring later in this section.

Using this information, automation planners can add and re-balance resources as necessary to meet the requirements. This may mean bringing additional resources online as demands dictate. It may mean moving jobs from overburdened servers to servers with additional capacity.

A system that simplifies locating where jobs run can make it much simpler to perform this balancing act. If jobs can be deployed from a single console, Operations can make changes simply and reliably. It will help them be more responsive and help them keep the system well tuned. Making the changes from a central source also makes the changes easier to audit and track.

Most IT departments are trying to do more work with the existing server resources that they operate. Systems that assist with tracking and reporting the utilization of IT departments' resources help them plan to maximize the utilization of those resources. Systems that aid in moving tasks between servers efficiently help IT departments implement those plans quickly and help the enterprise receive the best return on their IT investments.

### ***Prioritizing the Workload***

The IT automation workload tasks compete for resources, so the automation planners must have the means to prioritize plans. This is not a function of the technology so much as an understanding of the needs of the business.

Workload prioritization is often as political as it is practical. Every individual business owner feels that their work is the most important and can state a compelling case why it should be completed before other tasks. IT will receive pressure from many business sponsors to keep their jobs running on time and assuring their reports and processes are completed on time.

Some plans have practical implications. For instance, if the orders need to be batched so that orders can be shipped tomorrow morning and the shipping staff knows what they need to do, that job has some very practical priority. Although users do not depend on data backups, keeping the data secured is critical. The system must have time to perform the backup (preferably when it is doing little else) to ensure the continuity of the business information.

The IT automation and job scheduling system cannot directly address the question of whether it is more important to provide the CEO a dashboard by 7:00 AM each morning or provide the COO with a manufacturing schedule for each month. It can help address the reality that the priorities will shift. A system that can allow jobs to be re-ordered with minimal disruption will help Operations keep pace with the changes.

The other way of helping manage shifting priorities is making the best use of the server resources. Manually scheduled systems, or systems that are only partially integrated, often schedule additional time for tasks to accommodate working with systems to which they have no direct connection. This can cause delays in processing and prevent the organization from making the most efficient use of their resources.

Prioritization of plans is a matter of mixing the political with the practical. The priorities will shift with changes in business operations and the mandates of those in charge. Developing a system that makes the most capacity available can help get more jobs done with the existing resources. A system that can reorder jobs to meet the current set of priorities with minimal effort can keep jobs running and satisfy the requirements of the business.

### ***Working with Job Interfaces***

Once more than one system is involved, the information passed between systems must be handled through some type of interface. Creating interfaces between systems is always a bit of a challenge. Maintaining them when they break can be even more difficult.

There are several common interfaces used when systems share information. The oldest and one of the most common is sharing files. One system generates a file with data in it. That file is transported to another system that reads the file and uses its contents.

It would appear that once a file format is defined, this should be a simple interface to maintain. The common format may be character delimited files, fixed length data files, XML files, and so on. The source system produces the file, using the specification, and the receiving system opens the file and finds the data it needs. Most people who have worked on file transfer systems have encountered the problems that attend this scheme. Files are created incorrectly by the source system. It is missing a delimiting character. It is missing end-of-line characters or end-of-file markers. The receiving system cannot find the data for which it searches. And more often, the problem is that the file was not generated in the first place, so there is nothing to import.

The workload automation system can help by identifying the initial issue. It should identify whether the job failed to produce a file. Systems that track standard task logs will have this information without even logging into the source system. If the file was produced, it should help collect errors that would help the Operations staff identify the issue with the file.

Many job automation systems have their own internal mechanisms for exchanging information between jobs. As part of the automation system, it is maintained by the automation system. Such interfaces provide clear monitoring information and simplify the task of connecting one server to another.

Other programmatic interfaces are also commonly available. Many systems offer Web services to help share data. Remote connections through database libraries, ODBC, JDBC, and OLE DB allow communications with databases directly. Enterprise Java Bean applications and Microsoft COM and DCOM allow remote calls to existing functions. These mechanisms can make it simple to share information.

However, programmatic interfaces are also affected by upgrades to the systems that support them. New database client libraries, upgrades to Java runtimes, or patches to Microsoft COM or DCOM dynamic link libraries (DLLs) can cause unexpected side effects. Automation systems that alert the operation staff to these failures and help pinpoint the area that failed will keep the system running on track.

## Dealing with Security

Jobs may access data through programmatic interfaces. Web service interfaces, message queues, and database connections all require properly submitted security credentials to operate. There are several dimensions to security credentials required to keep plans operating. First, there is a wide variety of credentials required to access different services on different platforms: It can be as simple as a clear-text username and password; it can require logging the work thread into a Lightweight Directory Access Protocol (LDAP) service and attaching a security token to the identity of the thread; it may necessitate logging into Active Directory (AD) as a Windows user; or it might require presenting a security certificate to a server or network firewall.

Then the credentials need to be kept securely. The storage of the credentials is exacerbated because the credentials to execute the jobs often carry higher-level privileges to access data. The operators who oversee the jobs often do not have this level of privilege. Someone needs to be able to maintain the password without having elevated privileges to access the processes.

Systems that can delegate the security for running job tasks on their target systems can provide additional security. If they can provide security tokens rather than actually managing users' names and passwords, they can make such operations even more secure.

The IT automation system should be able to handle the wide range of credentials used within the organization. It should secure those credentials and allow them to be maintained easily. It should also help identify when programmatic interfaces fail because the credentials have changed.

### **Monitoring the Workloads**

As previously mentioned, a clear, responsive monitoring system is the key to executing schedules. The challenges in monitoring processes in an environment with multiple servers, hosted on different operating platforms, scattered in different facilities are easy to underestimate.

When a single plan consists of distinct tasks performed on different servers, it can be difficult to ensure that all the steps were completed. The plan is not complete unless all the individual steps are completed in the proper order, so there must be the means to track the jobs.

The system should also help Operations troubleshoot jobs. If one of the jobs in a plan fails, the automation system should quickly identify the step that failed. If the jobs provide proper error reporting and logging, this will help Operations get to the source of the trouble and correct it.

Server utilization is critical to understanding why some jobs start running late, so the performance of the individual servers must also be considered. An IT automation workload system that can integrate with the server monitoring software can help draw together all the information that Operations requires to understand issues and make good decisions about correcting the problems. An automation system that can automatically recalculate the expected length of the job execution can help Operations maintain the schedule with less additional effort.

The system can also proactively alert Operations of failed jobs. By receiving notification promptly, Operations can quickly make adjustments to keep the jobs on track and help the plan complete as close to on schedule as possible.

Some automation plans require auditing to guarantee that the system complies with the design standards. The monitoring system can validate that the plans are executed as designed. It can provide reports to demonstrate that the constraints, the regulation or policy that drives the job are being observed.

The automation system should also provide performance monitoring. By tracking how long the individual jobs and the integration of the jobs take to complete, it can provide baselines for determining where the schedule needs to be adjusted. It can help the IT staff proactively determine where servers are reaching their capacity. It can help find the servers with untapped capacity. It can also help justify the need to bring additional servers online.

## Overseeing Plan Orchestrations

A plan requires the movement of information. The systems that move the information form, what can be called, the orchestration of the plan. The difficulty in working with orchestrations can be that the systems that actually move information can be difficult to trace.

If the jobs use programmatic interfaces, such as Web services, CORBA, DCOM, .NET remoting, WCF, or other similar interfaces, the issue will usually be visible as a failure in the connection. This will invoke some type of logged error.

Message queues and enterprise service bus applications can be more difficult to track and troubleshoot. Items can be placed successfully into the message queue but not arrive at their destination. The arrival typically triggers the next step, so no overt error occurs. One must check the job queue to find the sequestered message and determine why it did not move. Often nothing in this process created an overt error, so only the fact that the job never completed would indicate that an error occurred.

Moving files can be difficult to track. Files placed in locations to be moved by FTP servers or mail servers can get lost. It can be difficult to discern whether the breakdown occurred at the sender or the receiver, particularly if neither logs an error.

An IT automation workload system watches the flow of the individual jobs from beginning to end. It can identify how far a plan moved through its composite jobs and determine where in the chain the plan stopped executing. Thus, even when the task generates no obvious error, Operations still knows where to look to troubleshoot the process.

## Time-Based Scheduling

The oldest and most easily understood paradigm for scheduling plans is time-based scheduling. The operator simply sets the server to perform a specific job at a specific time. When the time arrives, the job begins and runs until it is done. Later, another job will start when it is scheduled and so on.

The art of time-based scheduling is to use the time to one's best advantage. To begin to plan, one must understand the time constraints around which the schedule must be built. This includes the time windows available for processing, the order and deadlines in which tasks must be completed, and other constraints. The business calendar must also be considered. Week-end and month-end loads, seasonality of work to be processed, must be entered into the scheduling agenda. Constraint on resources, such as server and network availability, play a role. Synchronous and asynchronous tasks will also affect the operation of the schedule.

### **Working Within the Time Constraints**

When building the schedule of jobs, there are a number of factors that come into play:

- **Processing window:** For most servers, there is a window allocated for batch tasks. This window serves a dual purpose. It is expensive to keep people who are being paid by the hour waiting for a server to respond. Also, many batch processes run better when there are not a large number of people changing the data that the process is mulling over or backing up. The automation tasks will typically need to fit into that batch processing window.
- **Task processing time:** Each job will take some time to process. If that time were fixed, scheduling would be relatively easy. But most jobs will vary day to day, based on the data being handled. Data varies with seasons, weekly and monthly cycles, sales deadlines, and a wide variety of other factors. The scheduling team must allow for these variations in the job time to keep the schedule intact.
- **Time zones:** With enterprises operating throughout the world, the processing window for different data centers will vary. The schedule for the plan must account for the fact that some of the tasks for the job will be processed at 03:00 GMT, while others will process at 11:30 GMT. Scheduling a job will require spanning the time zone issues and knitting the output from the various tasks together in a timely manner.

The key to maintaining the schedule is to understand the requirements of that schedule. It requires a clear set of parameters and the means to measure those parameters. Many of those parameters were based on assumptions made by the automation scheduler when he or she originally established the plan.

The first consideration is the beginning and ending of the scheduling window. The window is controlled by many factors. Most often, it is determined by when human users access the system and when the bulk of its interactive tasks are performed. This is typically a bad time to run automation jobs (of course, if the server is available and in use  $24 \times 7$ , the jobs will need to share the resources). Some jobs need to wait until applications complete internal processes and the data is ready to be processed by other systems. Many systems have “legacy”-based windows. These windows assume that the end of the business day is 7:00 PM (or some other time) and then begin their batch processes. The time can be arbitrary and not based on the events occurring within the business or on the server.

Next, the scheduler considers the length of time that the job takes to complete and the resources that the task will consume. In order to create a schedule of sequential tasks, the scheduler needs to know not only when a task will begin but also when it will end. The scheduler will allow the task a window of time in which to operate, and typically allow some additional time to accommodate growth or unpredicted changes in server performance. That slack time can be a costly waste of available resources.

Operations staff needs a clear record of these parameters. They become the basis for determining whether the schedule is being met. Without question, the assumptions used to create the original schedule will change over time. By comparing the original plan to the “ground truth” of the system, Operations can determine when things need to be adjusted, and communicate those needs to the automation planners.

The IT automation workload system needs to help measure and report on those parameters to help Operations understand the “ground truth” of what is occurring. By configuring the monitoring to report in a manner that makes it simple to compare the design with reality, they can clearly see problems as the problems slowly grow and proactively adjust to prevent larger problems.

IT automation systems that work with server monitoring systems can help gather relevant data on the performance of the servers that links with the schedule. Sometimes the best course is to correct the performance of the server rather than adjusting the schedule. The systems can work hand in hand to provide Operations with the necessary information to find the correct solution.

### ***Working with the Business Calendar***

It is complicated to create a job schedule that runs each day. It becomes even more difficult to run tasks that run only once a week, once a month, once a quarter, or once a year. Yet businesses run on a calendar, and many tasks are affected by that calendar. It changes the frequency of some types of tasks and requires Operations to carefully monitor and adjust internal job execution accordingly. Jobs are affected by many different calendar events:

- Weekends often provide time for servers to execute tasks with fewer users competing for resources. They also often call for jobs that run to summarize the week’s activities and prepare reports and data for the next week.
- Daylight Savings Time (DST) can wreak havoc on a schedule if it is not planned for in advance. Losing an hour from a day can destroy that day’s schedule. Also, different countries switch back and forth from DST on different dates, so the adjustments must be made for each distinct set of DST rules represented within the organization.
- Most accounting systems will reconcile the end of the month during the first week of the following month. This will be accompanied by a number of ancillary systems that use the month-end reconciliation to update themselves. Thus, the servers incur extra work for a short period at the beginning of the month. Fiscal quarters and end-of-year reconciliations also change the workload and automation system requirements.
- Public holidays typically mean a day or two of reduced server activity, often followed by increased demand while the servers try to catch up from the work they did not do during the holiday (very similar to their users). For enterprises that span multiple countries and cultures, these holidays will occur at differing times on different servers. Regardless of the holiday schedule, plans must run on time to keep the business up-to-date.
- Seasons make a great deal of difference to many organizations. There are peak periods of activity followed by time when activity is reduced.
- Scheduled server maintenance will require servers to be shut down for a variety of maintenance tasks, from software patches to hardware upgrades. Just because the server is not available does not mean that the tasks it completes can go undone.

The business calendar can cause irregularities in the demand for computer resources. This increases the challenges to automation planners. But the calendar also provides great opportunity. There are two key elements to understand—job frequency and resource utilization. There are many jobs that do not run every day. From an Operations standpoint, these jobs are more difficult to track because they will affect the schedule less often. Simple scheduling systems that cannot adjust for weekend, holidays, or other calendar events may generate spurious errors when the business calendar creates scheduling anomalies.

An automation system that understands jobs that run with differing schedules can help make sense of this. The reports and monitoring systems can help Operations remember why the schedule runs differently during the first week of the month or at the end of the fiscal quarter or year. It can help remember to run jobs on atypical schedules.

Automation planners will also benefit from systems that understand the fiscal calendar and that can perform data arithmetic based upon it. If the schedule can be set to run 5 business days after the end of the fiscal quarter, the system will automatically adjust for other irregularities (such as holidays). This relieves the automation planners of the burden of determining this date manually.

The other opportunity involves the use of resources. The use of server resources is not uniform, so the availability of resources can be tailored to fit the need of the workload. This can save energy costs and reduce the load on IT when servers that are not required are not in service. With the advent of server virtualization, server images can be brought up to handle the extra workload and then archived until required again. Physical servers can be powered up and then shut down when their tasks are completed.

To support this type of flexible use of resource, the IT automation system needs to provide flexible use of server resources. The system needs to work correctly when servers come online to execute their individual jobs and then go offline when not required. The ability to move jobs simply from one server instance to another is also quite useful. If the automation system can work with a flexible infrastructure, it can allow IT to optimize the servers on which the tasks are performed.

### ***Adapting to Resource Constraints***

Jobs run for IT automation use a variety of enterprise resources: servers to execute individual jobs, networks to move data, Storage Area Networks (SANs) to store and retrieve data, and other similar resources. Most of these resources serve many demands. Thus, the workload placed on the resources never remains constant.

As Operations monitors the execution of the IT automation workload jobs, they will see variations in the time individual tasks take to run. These variations may arise for different volumes of data, but they may also arise from other processes that consume disk I/O, CPU time, network bandwidth, server memory, or other shared resources.

For instance, a database server may be set to recalculate statistics when a certain threshold is reached within the database. A data extraction job may cross that threshold and trigger the process of recalculating the database statistics. The drain in resources causes the job to run 30% slower. But the issue occurs intermittently. It may run another 3 days before it triggers the statistics process again.

The issue is not limited to servers. A system can be set to replicate directory data. If the directory replication occurs across a relatively low-bandwidth WAN link at the same time that a data file is moved across the same WAN link, the job will suddenly run much slower.

Because automation tasks share resources with a number of other jobs and processes that have no direct connection with those jobs, it is important to monitor the entire enterprise landscape. Any process can affect the schedule and the responsiveness of the resources used by the automation system.

There are many products that monitor the condition of enterprise computing resources. An IT automation system that works in conjunction with the monitoring system used by an enterprise can help correlate the health and performance of servers, routers, network links, SANs, and other shared devices with the activities performed by the automation system tasks. This can help alert the Operations staff of specific problems. It can also help the staff to determine when resources are in short supply. This information can help Operations re-balance the schedule and make adjustments to keep the automation schedule running on time.

Resources will also go offline. A server might crash or be taken offline for maintenance. A network link may fail. Operations staff needs to know when resources are unavailable. They then need the means to quickly change the way that the job tasks execute so that the job itself is still executed as required.

An IT automation system that allows jobs to be quickly and reliably re-assigned to different servers or network resources can help Operations make adjustments and help the plans keep to their schedule.

### ***Running Tasks Synchronously vs. Asynchronously***

Most of the resources on which the IT automation schedule executes are made to multitask. They can run multiple jobs at the same time. Of course, the more jobs that a server runs at once, the slower each job runs individually. When multiple jobs are run on the server, the Operations team must understand the overall effect this will play on performance. Once again, linking the IT automation workload system to the performance monitoring systems can help Operations know what is happening when their jobs execute.

Beyond the simple loading of jobs on the server at the same time, there are some jobs that run as pre-cursors to others. Some job tasks must run in a specific order on a given server in order for the jobs to run correctly or securely. Other jobs must run first in order to deliver output to another system in a timely manner.

For instance, a server might need to run a reconciliation to summarize the day's shipping activities. Before the reconciliation runs, IT may mandate that the system is backed up (in case the reconciliation fails and the state of the server must be restored). Obviously, the backup must occur before the reconciliation can begin. These jobs must be clearly defined as serial tasks so that, when Operations needs to adjust the schedule, the jobs do not end up out of order.

Conversely, two data extracts may be able to run in parallel. If the server has the capacity to query the database for multiple results without a serious reduction in response, these jobs can be run side by side and complete the tasks in a reasonable period. Running jobs in parallel can help take full advantage of the capacity of the system.

Automation planners need to know which jobs run serially and which jobs run in parallel. Operations needs a system that will help track the performance and let them know when the schedule no longer meets the design specification determined by the planners. Automation planning and Operations need to work together to determine which jobs can be moved. By working together, they can optimize the workload across the enterprise servers. A flexible system that helps Operations re-deploy the job tasks will make this much easier, more reliable, and cost effective.

Jobs running on one system will affect the jobs that follow. When these jobs occur on other servers, it can be more difficult to understand the impact of re-scheduling plans. For instance, a data extract may be performed on the mainframe. It does not affect any other jobs on that server, so it may seem easy to move that data extract back in the schedule. But that data extract may be required by the ETL system to move to the data warehouse. The data may need to go through several processes to be included in the Online Analytical Processing (OLAP) database that is used to report the daily production activity. The data warehousing processes may need the data extract early in the evening, even though it makes no direct difference to the mainframe system when the job processes.

An IT automation system that helps Operations visualize the interconnections between automation jobs will help reveal how they need to be scheduled across systems. Often, the IT automation system will be the only system that clearly shows the relationships of these jobs across the enterprise. Being able to adjust the schedule of those plans on multiple servers can save a great deal of time and money for the enterprise.

## Event Driven Scheduling

Event-driven scheduling is a common programming paradigm, developed shortly after the time that computer programmers determined that computer resources should be invoked when people needed them, not necessarily when the clock said it was time to work on them. It is uncommon to think of automation schedules to be driven by events, rather than time, but event-driven scheduling can be a very effective means of maximizing the utilization of computer resources.

Event-driven scheduling starts by using a different set of interfaces than are typically used by batch processes. These interfaces need to be monitored using different tools. Events can occur at inexact times, so the balancing of resources may take on a different set of considerations. Monitoring of events also poses a distinct set of issues than those of more traditional time-based schedules.

The events driven by these systems are often external events, not in control of the servers that perform the jobs of the automation system. File triggers and Web-Based Event Management (WBEM) triggers are fired. The job will remain dormant but ready to react when these events occur.

### **Working with On-Demand Interfaces**

There are a variety of on-demand interfaces that can be used to access programs and services within the enterprise. These interfaces allow systems to communicate with one another. Automation jobs can be joined into contiguous plans by using these interfaces to share information.

Programs in many platforms (Java, C++, Microsoft COM, and so on) allow one process to call to another. These paradigms have been extended to call across machine boundaries. Java Enterprise Beans applications, CORBA, Microsoft DCOM, and Windows Communication Foundation are all examples of these types of interfaces in practice.

A system can initiate the job externally as well. For instance, an order entry system from customers may collect orders and build a consolidated order file. When the file is prepared, it uses an FTP server to pass the job to an order fulfillment service. The order fulfillment service begins a set of automation plans whenever the receiving file system receives the order file.

Calling between processes can be hindered when calling across platforms. Simple Object Access Protocol (SOAP) and Web services help processes built in one platform to call objects built in another platform. It helps diverse platforms interact.

Another common type of automation interface is a message queue. A job can execute and place its results in the queue. The queue can translate the message from one format to another (for example, convert a CSV file into an XML document) and then move it to a receiving application. The message can be held until the receiving application is able to “pick up” the message and process it.

These programmatic interfaces offer new opportunities to orchestrate automation tasks. One job can call the next whenever it is prepared—no longer on a synchronized clock schedule. This can make the entire plan flow rapidly from one completed job to the next.

Monitoring and troubleshooting programmatic interfaces requires the IT automation system to look for issues and problems. When programmatic interfaces fail, they often log issues in different places than standard batch jobs. When a call fails, there needs to be different mechanisms to determine the source of the failure—the calling or the called process. The calls will often require security credentials. The credentials need to be kept current, and security failures are often at the root of problems.

IT automation workload systems that are aware of programmatic interfaces are equipped to help monitor their activity and locate the source of trouble. They should be able to track when programmatic calls were made and the result of these calls. This helps to abstract the chain of events as one job calls the next. It can be used to create reports to validate that processes have run correctly and to trigger alerts when processes fail.

Interfaces such as Web services and message queues are often handled by farms of servers. Calling a programmatic interface on a farm can be difficult to audit. An IT automation system that conceptualizes the call to the service as a job and does not require the job to be a process run on a single server will monitor these interfaces more accurately.

### ***Dynamic Utilization of Resources***

One of the characteristics of on-demand jobs is that they are more challenging to audit. If a job is scheduled, a monitor process can expect the job to run and then, using its own independent clock, determine whether it ran. With on-demand resources, it may be normal for a particular job not to run. It may also indicate that there is an unreported issue with the calling job. Tracking plans will need to be performed using different means.

Another issue is the utilization of server resources. With a time-scheduled task, the Operations staff knows when a particular job will run. They are able to plan the load on the server at that time and ensure that the resources are available to execute it in a timely manner. When servers support on-demand jobs, there is no way to firmly establish exactly when the job will run. This can make balancing the load much more challenging.

To manage on-demand resources, the automation planning staff needs some type of expectation of when the jobs will run. The reason that batch jobs are not run during peak business hours is so that the servers can respond quickly to the needs of users. The servers are sized to meet the peak expected demand. Servers often have some idle capacity so that they have sufficient capacity to meet those peak periods. If the IT automation system can add to the demand in an “as required” manner, additional server capacity will need to be made available to handle that load as well.

To support this, the IT automation system should have some expectation of when the jobs will execute, even though they run on demand. The Operations staff can use this expectation to size the servers appropriately and plan workloads that the servers can accommodate. This may mean that the IT automation system will be required to contain automation plans that run on demand to limit themselves to a specific window of opportunity. Thus, the task will run as required, but only during selected hours of the day.

IT automation systems that simplify the process of placing jobs on the appropriate servers can make it much simpler to adjust conditions to ensure that all the processes that the servers handle can get the resources that they need.

### ***Monitoring On-Demand Job Tasks***

Monitoring jobs that run on demand can be challenging. The design of the tasks must provide some definition of the expectation of when plans are likely to run. This allows the expected available resources on the server to be projected. It also allows a monitoring schedule to be established.

There need to be parameters set for monitoring that indicate how long the plan can go without running before Operations begins to look for errors. If the plan does not run within that maximum window, the reason it has not run should be investigated. Not all job failures generate clear errors, so this investigation will prevent jobs from falling between the cracks and being left unexecuted.

An example of how this helps can be seen in message queues. Sometimes a message is successfully entered into the queue, but it is never picked up by the receiving process. These can be some of the most difficult problems to locate and resolve because there is not a specific error generated. The IT automation system may independently monitor the receiving process. If the receiving process is monitored and does not activate within a prescribed period, it will send an alert to the Operations staff. The alerts will help the staff stay on top of unexpected occurrences that do not necessarily appear as errors.

Operations will also need to carefully monitor the affect of the automation jobs on the server resources. The best way to utilize a server in an on-demand environment is to allow it to serve other purposes while it awaits its next call, so the way that these varied tasks compete for resources will determine the performance of the automation systems and the other tasks for which the server is slated. If the IT automation system can help to correlate its demand with the enterprise performance monitoring systems, this correlation will help Operations to see how these tasks affect one another and correct any conflicts that may occur.

Error detection with on-demand jobs also takes a different flavor. Some errors will be reported to centralized, monitored repositories, such as event logs, databases, and the like. But many on-demand tasks will include the error in a message that is returned to the calling process. It may not be reported anywhere else.

For instance, many Web services will create a section in the XML payload that contains any errors. The Web service itself may make no other notification of the error. It becomes the responsibility of the calling process to handle the error. If the calling system does not notify something else that an error occurred, the plan will stall at this step. It can be very difficult to determine what is occurring if these messages are not placed somewhere that they can be tracked.

The on-demand jobs need to be configured to notify the system when an error occurs and provide the contents of the error message. This may require structuring the calling job to format and send the error to a monitored repository. An IT automation system that can receive, interpret, and help report such errors can help Operations monitor the operation of the jobs. If the message content can be provided, that information also provides the means for IT to begin to troubleshoot the issue and make any necessary corrections.

## Job Execution

Somehow, things that work very well in design and test mysteriously fail to run the same way in production. The hardware, networks, and procedures used to control production are typically better and more carefully monitored. Nonetheless, issues seem to present themselves more readily there. And problems in production will ultimately affect the overall success of the execution schedule.

The task automation system must be equipped to handle changes in the real world systems in which they operate. They must be able to handle diverse topologies, such as clusters, Web farms, and distributed processing systems. They must adapt when tasks are moved from one server to another—as servers are consolidated, virtualized, or scaled out. The systems must have the ability to balance load across available resources and adapt to component failures without failing to complete their designed missions.

### **Working with Server Topologies**

Many organizations found an entry into computing by leasing time on a single mainframe. They had but one computer with one way of scheduling plans. Now organizations have dozens, hundreds, and even thousands of servers. In order to make jobs manageable and keep costs contained, the servers work together in farms and clusters. Jobs are moved across large numbers of different boxes that work together to complete a comprehensive service.

Many systems now use farms or grids of servers to scale out and handle the load. Rather than invoking a task on a single box, the job is submitted to the group. Any given server in the group can handle any portion of the requisite work.

When working in server farms, the service needs to be seen as the target system, not a given server. Often, requests can be made in parallel and the work will automatically be spread among the servers within the farm. When the farm has too much work, additional servers can be added. When they have less work, servers can be removed.

This provides some unique challenges for Operations. It is relatively simple to create schedules when time and resources are fixed. If the resources fluctuate, the system will run more inconsistently. Operations must carefully monitor the farm as a whole.

Some adjustments to jobs may be required. When working with application server farms, it is often more effective to submit multiple requests for smaller pieces of work. Such requests can be distributed among several servers thus using their resources. A single, long-running job may run on a single server, leaving the other servers idle.

An IT automation system that makes task re-configuration simple and reliable to accomplish may help re-configure the workload to make full use of the available resources. Also, if the automation system is aware of the cluster, it may be better suited to monitor the tasks submitted to the group rather than trying to track individual tasks executed on individual servers.

Clusters can provide a challenge as well. A failover cluster will have one server actively executing the workload. If that server fails, another server will assume the responsibility of executing the workload. The cluster may failover due to a system failure or failover manually so that IT can perform maintenance on the server.

The IT automation system needs to work seamlessly with the cluster system. Some failovers occur transparently and seamlessly. Other systems may fail over manually and require IT to reconfigure the components that use the server. Automation systems that are designed to work with the cluster will make failover simple and help Operations keep the tasks running on schedule.

## Changing with the Environment

A truth of enterprise server infrastructure is that it is in constant flux. For an enterprise to remain nimble and responsive, that enterprise needs to place workload on the servers where they have available capacity. There are many reasons for this type of change:

- Server consolidation can reduce hardware and power consumption. It can save precious space in the data center, reduce server licenses, and reduce the cost of keeping servers patched and running. It can simplify management. Organizations that can minimize the number of servers that they use can realize significant cost savings.
- Server virtualization offers tremendous potential for improved manageability and cost savings. Reducing the number of physical servers operating can save money. Virtual servers can be added quickly as demand requires, and then taken offline when not needed. Distinct server images can host specific applications and reduce the potential conflicts when a server hosts multiple applications.
- Corporate expansions, mergers, and acquisitions can add and remove servers from the enterprise. It can change where servers are located and require new IT automation and integration jobs to be added quickly. In addition to server locations, network connections can change. This can alter the load placed on WAN links and affect the overall performance of plans that use those links.
- Applications change and migrate. That migration can create new plans or eliminate long-standing plans. Directory services can change, altering the security requirements of the processes that those services help secure. Even simple application upgrades can call for the alteration and re-configuration of the tasks those applications support.

IT can act as a key advantage or a liability in these changes. If the organization can quickly and proactively manage its internal infrastructure, it can provide a higher level of service to the business, often at lower cost. If it can make changes accurately and reliably, it can help make corporate transitions easier.

For example, if an organization merges with another entity, one of the greatest challenges is merging the IT systems and infrastructure. An organization that can make the change quickly with fewer errors will be able to gain more competitive advantage for less cost. Those mergers often involve job automation to help the systems work together, so a responsive job automation system that can deploy jobs quickly on different platforms can help bridge the gap between the organizations and help them communicate sooner.

With all this potential for change, the IT automation system needs to be flexible. It must support a wide range of industry platforms. The system should allow Operations to re-configure plans in quick, reliable, and repeatable manner. If the automation schedule can be re-configured, tested, and validated quickly, IT can remain responsive to changing needs. That responsiveness can help contain IT costs and let the business take advantage of new opportunities.

### **Load Balancing**

There is often more than one way to complete a task. When creating schedules, the automation planner must consider how to balance the tasks across the available resource. Sometimes the resources vary, and sometimes the workload varies. Systems that can deliver the right workload to the right system at the right time offer the best opportunity to make full use of the enterprise resources.

If resources are unavailable—either from scheduled maintenance or other short-term, unexpected events—it is helpful to find alternative means for executing the workload and keeping operations on track. If alternative means of accomplishing jobs can be devised and implemented quickly and cost effectively, the schedule can be maintained.

There are, as previously mentioned, variations in the workload itself. As the business calendar varies, the number and variety of jobs will change, often day to day. If the plan schedules can be designed to allow for these variances and move work to servers that can handle the load, the jobs can be accomplished and the best return from the IT investments realized.

One of the easiest means of accomplishing this is by implementing conditional logic in the plan schedules. An IT automation system that can use variables to determine where and how to execute the jobs can be fine-tuned to automatically get work done where it will best be completed. The variables can be set by a variety of sources. It may be set by performance monitor values or by time or events within the business calendar. It may be set manually by the Operations staff. Having a system that can change its pattern of task execution by merely setting a parameter can make the system easy to control in changing conditions.

### **Error Correction and Compensation**

Almost all systems will encounter occasional errors. But the fact that the system failed to execute does not relieve Operations of the responsibility of getting the plan running quickly and restoring operations back to normal. The error must be identified. Preferably, the error can be corrected quickly and the plan re-run to restore order. Sometimes, the plan can be configured to correct itself and re-run, thus compensating for the error. If the error cannot be quickly corrected, the Operations staff may need an alternative means of accomplishing the plan, if even temporarily.

For instance, a scheduled task might need to pick up a file produced by another task. If the file is not ready, some plans can be automatically re-scheduled to try again in a few minutes. If the first job just ran late, this re-scheduled pickup will keep the plan running.

When plans are very critical, it may be worthwhile to design an alternative means of completing the plan. If a server fails, there may be another server or process that can accomplish a similar job. It may not be the preferred method but designed to execute only if the preferred method cannot run.

IT automation systems that support conditional logic can be more easily designed to work using alternative methods. They may run a job on an alternative server if the primary server is unavailable. They may choose to use a more expensive communication channel if the primary channel goes down. This type of intelligent design can make a more stable and reliable automation schedule.

Sometimes, there may be a task for which there is no viable alternative. When it fails, it simply must be corrected. The IT automation system can help to identify the miscreant job. It can provide troubleshooting information to help Operations get quickly to the heart of the problem and remediate it. The system can then help by re-running the plan on demand, communicating with all the servers related to the plan to ensure that the remediation was successful, and see to it that the plan is completed as quickly as possible.

Automation systems that can automate these restart tasks can save time and keep the system reliable. Systems that handle failovers without manual intervention will provide higher levels of service and minimize the loss of enterprise computer resources and time.

## Monitoring IT Automation Workloads

The importance of monitoring has been mentioned over and over again in this chapter. When overseeing operations on multiple servers, often located in different data centers, it becomes difficult to track plans running with a wide variety of schedules. Operations staff needs a centralized repository of information that can help them understand the execution of automated job streams within the organization.

The IT automation workload system should provide alerts to help Operations quickly respond to interruptions in the plans. It should also provide a reporting capability to help demonstrate the function of the system and provide an analytical basis for maintaining the schedule. The system should also help IT demonstrate their compliance with regulatory and corporate standards for handling data and executing processes.

### *Alerting Functions*

It is always helpful to receive an alert when things go wrong. If Operations needs to monitor every operation to determine that it ran correctly, errors will occasionally be missed. The more plans that need to be monitored, the more information that the Operations team needs to review, the more difficult it is to spot trouble. It becomes more difficult when plans are scattered on multiple servers with different means of reporting progress.

The IT automation system can monitor the specific activities of plans. When one job completes, the next job in the plan should start. The IT automation system can monitor activities from different servers than those executing the jobs, so the monitoring system will operate even if the server executing the job fails.

The automation system can be configured with alerts that can notify operators when jobs fail or the schedule is not being satisfied. Systems that provide relevant information can help Operations troubleshoot and find and correct the error quickly to keep the job stream flowing.

Not all plans have regular schedules, so the alert functionality should also monitor when plans have not run for a long period of time. Setting timers on plans that reset automatically when the plan executes can help remind IT that plans that run on irregular schedules have not run at all. This can help prevent these types of plans from being neglected.

A system that provides a variety of alerting mechanisms can help reach Operations under a wide variety of circumstances. Console notification can be used if consoles are monitored continuously. Emails and pager functions can be used to reach Operations staff that is away from the console.

### **Reporting**

Operations staff needs a great deal of information about the system to keep it running effectively. They need to know how long individual jobs take to run. They need to know when the jobs run so that staff can balance resources. They need to be able to validate that jobs have been run. They need to know when changes have occurred.

An IT automation workload system should provide a repository of information that the staff can use to determine this information. It should provide specific details about when, where, and how individual jobs are running. It should be able to correlate the individual jobs that constitute a plan and report on the operation of that plan as an entity. It should provide a basis for performing analysis of the operations. These analyses can help project future needs and provide the factual basis for making proactive changes to the way that the schedule is executed. The reporting system should also contain enough forensic information to determine why jobs fail and how those failures can be corrected.

An IT automation system that offers a flexible reporting structure can help generate reports that impart the proper insight into the way in which the automation workloads are executing. It should provide standard reports that help clearly expose the reality of how the workloads are running. It should also offer flexibility to help organizations create custom reports that provide the precise information that the organization requires to make good decisions.

### **Auditing**

Compliance has become a vital consideration in IT operations. Government regulations have required businesses to demonstrate that they handle and secure data in an appropriate manner. Regulations such as the Health Insurance Portability and Accountability Act (HIPAA) and Payment Card Industry (PCI) data management require that personal information remains protected and encrypted at all times. The Sarbanes-Oxley (SOX) Act and other regulations require that IT operations comply with specific standards and that the enterprise can prove that it met these requirements. Corporations increasingly require careful auditing of who accesses data and where that data is used and stored.

Auditing becomes an important part of any IT operation. Many IT automation tasks will involve moving sensitive or confidential information, so there must be a means of ensuring that the data was handled in an appropriate manner. There must be evidence that certain processes were in fact run, and a record of when they were run. Changes to the time and manner in which plans are run often need to be recorded and made available to audit.

Auditing is tedious work that is very simple to overlook and allowed to fall by the wayside. Incorporating many of the mundane tasks of auditing into the automation tasks can relieve Operations of low-level work and free them for more important tasks.

The IT automation system should provide assistance in this auditing. The system should not only execute the individual jobs in a compliant manner but also provide reports that validate that the operations have met the standards established by the governing bodies and by the corporation.

## Summary

Keeping a schedule of plans running day in and day out can be quite challenging. When mired in the daily issues of keeping individual jobs running, it can be difficult to keep the end goal in focus. When changes occur in enterprise infrastructure, it can be challenging to keep the scheduled plans running. It can be hard to monitor the systems and find where the best place to run a job is hiding within the enterprise.

An IT automation system that provides flexible support can help overcome these difficulties. A system that helps visualize and plan individual jobs as part of a single plan can help Operations see the interrelation of the jobs—even when those jobs execute in different locations on different servers. It can help find the optimal order in which jobs should run to satisfy all the needs of the organization.

An automation system that provides flexible deployment of plans can help re-arrange when and where plans run. Operations can target jobs to run in the most effective place. They can move jobs when the infrastructure changes. They can create alternative means of completing plans to accommodate for seasonality, maintenance, or other cyclical or unpredictable events.

Automation systems that can work with on-demand scheduling of jobs through programmatic interfaces such as message queues and Web services can open new opportunities for plan scheduling. They must also provide monitoring for these interfaces.

Automation systems that provide effective monitoring can alert Operations to trouble as soon as it is encountered. Such systems can provide reports that validate the operation of the system and offer an analytical basis for proactively managing the schedule. They can also help satisfy the auditing and compliance requirements placed on IT.

## Download Additional eBooks from Realtime Nexus!

Realtime Nexus—The Digital Library provides world-class expert resources that IT professionals depend on to learn about the newest technologies. If you found this eBook to be informative, we encourage you to download more of our industry-leading technology eBooks and video guides at Realtime Nexus. Please visit <http://nexus.realtimepublishers.com>.