

Realtime
publishers

"Leading the Conversation"

The Shortcut Guide[™] To



IT Workload Automation and Job Scheduling

sponsored by



Mark Scott

Chapter 4: Best Practices for IT Workload Automation and Job Scheduling61

Service Level Agreements62

 Performance Requirements62

 Resource Utilization.....63

 High Availability and Disaster Recovery64

 Notification, Monitoring, and Auditing Requirements65

 Notification65

 Monitoring66

 Auditing66

Job Deployment and Execution67

 System Skills and Training67

 Leveraging Enterprise Capabilities68

 Job Execution69

 Building Durable, Resilient Job Automation Systems69

Monitoring and Auditing70

 Correcting Errors70

 Validating Compliance71

 Capacity Planning71

Resource Management.....72

 Running in the Right Place at the Right Time72

 Server Power Management73

 Plan Configuration74

System Architecture.....75

 Growth and Adaptation.....75

 Scalability76

 Object-Oriented Designs.....77

 Architectures that Keep Plans Running on Schedule78

 Working with the Target Servers79

Summary80

Copyright Statement

© 2009 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.

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

Chapter 4: Best Practices for IT Workload Automation and Job Scheduling

When organizations approach solving their business challenges, they want to know what approach is likely to solve their problem. A proven solution can save time, money, and frustration when developing the solution. This chapter will address approaches to IT workload automation and job scheduling that have been vetted by success in many other organizations. It should provide a place to begin when solving the specific challenges for a given enterprise. It outlines topics of consideration that every organization should address to determine how to define, deploy, and manage their automations system. These topics of discussion include:

- **Service Level Agreements:** To build a system that meets the needs of the enterprise, IT and the business must agree to the level of service that they expect. The demands for performance and limits of budget must be balanced and an agreement reached so that everyone knows what to expect. A clear set of service level agreements (SLAs) provide the key ingredients for planning a successful workflow and job automation system.
- **Job Deployment and Execution:** The system must operate cost effectively. Beyond the purchase price of the system, organizations must consider the operational costs, maintenance costs, and efficient use of existing resources. Mapping out these costs can help design and implement a system with a higher return on investment and a system that will provide years of additional savings to the organization.
- **Monitoring and Auditing:** There is no way to operate or improve the system if there is no record of what the system is doing. A well-designed system provides timely notifications to system operators of issues and provides them with the right information to correct problems effectively. Integration with enterprise resource monitoring systems provides a more complete picture to allow automation planners to optimize the system and plan for the future. The ever-expanding regulation of corporate data and system operations also makes auditing a key component to successful automation systems.
- **Resource Management:** The automation system works on the corporate servers that provide information to the entire organization. Using those servers effectively can help reduce server counts and bandwidth constraints. From making the best use of licenses to opening up data center real estate to conserving power, systems that make optimum use of corporate resources reduce costs.
- **System Architecture:** Businesses grow, change, and adapt. The automation must take these changes in stride. By providing a nimble, resilient architecture, the automation system can help the organization make the best changes with little anxiety over the ability of the automation system to fulfill its mandated role.

Service Level Agreements

The business wants data moved from system to system instantly, for free. But there is a real cost to move data from point to point. The purpose of the SLA is to help express what the business will get with the budget that has been allocated for the service. That budget includes monetary expenditures, consumption of corporate resources, and time windows for operations. The key points to develop when building SLAs include performance requirements, resource utilization, high availability/disaster recovery options, and monitoring and auditing requirements.

Performance Requirements

The business needs to depend on the job automation system to deliver work in a timely manner. The performance requirements are meant to help define that timeframe. For a job automation system, the performance requirements will specify the time windows in which the jobs need to be completed.

For many organizations, there is a period of time during which the servers are not in high demand from the users. Typically, this is the period when day-end reconciliations are run, data is backed up, and reports are processed. There is often a window of opportunity during which the automation systems have better access to the servers to complete the automation tasks. This window of opportunity should be defined in the performance requirements to help the automation planners define the plan schedules.



For the purposes of this chapter, a plan is a series of one or more individual jobs run to accomplish a business goal. A job is an individual operation run on a server. Thus, if the business goal for a plan is to prepare a report on production for the previous day, the plan may include a job that extracts data from the production system, a job to import the extracted data into the data warehouse, and a job to process the report.

Automation systems may also be used to handle work on demand. This can be jobs that are triggered through the enterprise's event driven architecture or the use of Web services. When an event triggers a plan, the system will work with the enterprise servers to complete the operations. The users and service owner need to understand how these demand-driven plans will affect their services. They should help define when the plans can run and when they should wait.

Some jobs may need to be queued until resources are available, until other input files are available, or until other plans have been run that prepare for the job. The performance specification should define the allowable waiting period for a plan to be delayed. It may also define what happens to plans that cannot be run within the specified time period.

The automation planner must take the performance requirements and use them to develop a set of plans that will accomplish the stated business goals within the timeframes defined. That will require the planner to estimate the time to run the plans. It will also require the planner to see how the plans can be scheduled to run within the existing enterprise using the available resources.

The automation system should help the planner to design a schedule for the plans that will meet these requirements. A system that can show all the plans in a single point of control will help the planner see the way in which the plans affect one another. It will help him or her to orchestrate the plans so that each job can run in the available allocation and meet the expressed goals. Such a system can help the planner satisfy the performance requirements.

Resource Utilization

The automation system ultimately executes its tasks on servers that are used by others. It must share the capacity of the systems within the enterprise. Thus, the enterprise should understand the utilization required by the automation tasks and set that as expectations or constraints within the SLAs.

The resource utilization can be thought of in several ways. For instance, there may be a period during which user input will not be accepted so that reconciliation processes can be run. There may be a period during which data extracts will consume most of the disk access and the server will not be useful for any other purpose. When a Windows Management Interface (WMI) event triggers a data backup or cleanup plan, the servers will be diverted to accomplish those jobs.

There are times when automation systems need more resources. Fiscal period ends, seasonal order increases, maintenance cycles and other occasions can place extra load on the data systems. This may require planning to adjust for the load, or may call for additional server resources to be brought online.

Resource utilization will require coordination with the other enterprise operations. An SLA that denotes these requirements can help foster communications between departments and system owners. It can also help to provide the basis for server planning, both in server expansion and consolidation.

The automation system should have clear boundaries for the resources that it consumes. The business should make clear when jobs can and cannot be run. The constraints may be driven by the business processes themselves. For instance, month-end reconciliation cannot be run until the books for the month have been closed. A job that processes an external data feed cannot run until the feed is available. Other constraints have to do with the availability of the resource. If an automation job consumes 80% of the capacity of a server, it should not run during the peak time when users are trying to log on to the system. Some jobs may also interfere with one another if they are run on the same server at the same time. The automation planners need to share the resources while still keeping the plans operating on time.

The SLAs should help determine what resources can be made available to the automation planners and when they can use them. The planners then need a system that helps them coordinate the scheduling of the plans using these constraints. The plans may call for using a variety of servers, often located in different data centers to accomplish the tasks. Thus, beyond the use of the servers, the SLA should also consider the use of bandwidth connecting data centers. The impact of the automation system should be clearly communicated to everyone involved.

High Availability and Disaster Recovery

All systems are subject to failure. From natural disaster to component failures to mean-spirited hackers and malicious code, systems fail. Enterprises should plan for the eventuality of these failures and make plans to keep their organization operational. For the automation system, there are two types of failures to consider. The failure of the target systems on which the jobs run and the failure of the automation system itself.

High availability may be handled at the individual server cluster or farm level. If the servers themselves distribute the load, plans can submit jobs to the balanced endpoints. But many servers are not balanced for load sharing. For these systems, the planner may need to configure the plan to use conditional branching to choose the server that has the most capacity.

This is not limited to the server. Data centers may have multiple independent links for wide area networks (WANs). Automation systems that can work with SNMP events can properly route message traffic to move data as quickly and cost effectively as possible.

Regularly scheduled maintenance may take servers offline. A backup server is often pressed into service for this outage. It may be that two servers are rotated between the primary and backup roles to facilitate maintenance. A system that allows the abstraction of servers as objects can simplify the task of creating alternate versions of the plan that work with the available servers. This helps the enterprise plan for maintenance with minimal disruption.

When servers fail unexpectedly, plans may need to compensate for the lack of the server resource. The planner may need to design plans that have alternate solutions to keep things on schedule. A system that integrates with the server monitoring systems and that can use conditional branching lets the planner devise a plan that can take these unexpected events in stride.

A well-implemented automation system will handle failovers transparently. It will automatically restart jobs and keep the plans as close to schedule as possible. This type of automation helps minimize disruption and prevents the plans from adding to the list of woes incurred when a server fails.

Some plans should be run only in the event of an unexpected system failure. Using an event-driven architecture, a system failure can be used as a trigger to activate a plan. This may save vulnerable data or proactively execute jobs to help the enterprise cope with the loss of the server resource or resources.

The automation system itself runs on servers that may require maintenance or fail. An automation system that provides high availability will help ensure that the plans do not have a single point of failure. A distributed architecture on the automation servers can help reduce bandwidth requirements and make for a more durable system. It can also help design an automation system that can be protected from a data center failure. The SLA should define how quickly the system must recover from failures.

Good designs allow the automation process to continue even if a scheduling server fails. The plans should allow for automatic restarts. Checkpoints designed into the plan should show how far the plans have progressed and where to pick up the disrupted schedule.

Notification, Monitoring, and Auditing Requirements

The successful operation of the system will depend on the operations staff to monitor its activity. There are three areas that are important to consider in this realm: notification, monitoring, and auditing.

Notification



For the purposes of this discussion, notification is the process by which the system lets the operators know when a plan fails to execute.

Notification allows the staff to react quickly when problems are encountered. By letting the staff know as soon as possible, it affords them the opportunity to make corrections and keep the plan close to schedule. The more critical the plan is to the operation of the business, the more significant is the notification system. The business needs to determine how they will keep the staff informed of issues with the system.

The automation system can help by providing a full range of notification mechanisms. Some organizations will require pagers, others may use cell phones, others will rely on emails, and still others will have staff monitoring the console. Integration with another monitoring system, such as OpenView or System Center Operations Manager, may be preferred. An automation system that can meet the present and future requirements for notification will allow the business to choose the approach that best suits their needs rather than acting as a constraint. The SLA should define the mechanism that will be used for notification.

Monitoring

 For purposes of this discussion, monitoring can be considered the process of recording the activities of the automation system.

The system should provide detailed information about the execution of plans—those that succeed and those that fail. This information can be used for two purposes. One is to validate that the system is functioning as required and the other is to provide the baseline information to evaluate changes to the system.

Any automation system monitoring should provide validation that it is indeed operating. Automation systems often require only minimal human interaction. If they do not generate errors, they can be overlooked. This can present certain risks. For instance, a plan may execute but have no data to process. This is not necessarily an error, so the notification system may not be triggered. But if the operation staff is confirming why the data did not process, the monitoring system provides the information necessary to follow the process. Event-driven architectures may also not run because the triggering event did not initiate the plan, or the triggering event never occurred. Sometimes plans run correctly and another system fails. It is also affirming to see the plans ran as expected and that everything is as it should be.

The automation system should provide a central repository of data on its operation. The system should provide a full range of standard reports to present the operational information in a clear, easily understood manner. It is also helpful if custom reports can be built from the data store. This will help tune the information to the specific needs of the organization. The SLA should help define what reports are required to ensure the system has the requisite monitoring.

Auditing

 For purposes of this discussion, auditing is the collection and reporting of data to validate compliance with corporate and government regulations and standards.

Every enterprise is unique, thus the auditing required to ensure the compliance will also be unique. Different organizations will have different groups to which the audits must be presented. It may be government organizations, industry regulatory bodies, parent organizations, business partners, vendors, or clients.

The automation system needs to be able to provide a flexible means of collecting significant data and reporting that data in a manner that is acceptable to the auditors. The format may vary from simple paper-based reports to XML documents or other formatted files.

The SLA should define the reports required for auditing. This definition will help ensure that the automation system captures all the required data. It should also define the format in which these reports should be prepared and submitted.

Job Deployment and Execution

The automation system is deployed throughout the entire enterprise. It touches many different applications and platforms, often in many different locations. Deploying jobs and executing them in this diverse, changing environment is a major consideration when developing the system.

There are several things that need to be considered. The skills requirements and training of the operations staff is crucial. Making the best use of the existing enterprise services can help reduce maintenance and lower total cost. Providing control of systems from a central point eases systems management concerns. Designing resilient, self-correcting systems keeps the automation system running on schedule.

System Skills and Training

A plan can execute on many server platforms. A data extraction may take place on a mainframe. The data may be moved with an SFTP link to a UNIX server that imports the data into an Oracle database. Another Linux system creates a second file extract and moves it to a folder for import. Once the data has been imported into the Oracle database, it is then extracted into a Microsoft Analysis Services cube, and SQL Server Reporting Services generates subscription-based reports and emails them to the subscribers. This is a typical plan. It requires executing jobs on mainframes, UNIX servers, Linux servers, and Microsoft servers. It requires using email, SFTP, and network database connections. There are many different moving parts, all of which need to be carefully coordinated.

Automation planners need an automation system that can run on all these platforms. The system should provide a single, unified paradigm for running jobs and linking the jobs together to form plans. A system that can provide a simple, easily understood interface will help planners concentrate on the plan itself rather than the details of implementation.

Similarly, the operations personnel need to be able to monitor the execution of the plans and make adjustments. An interface that provides a single point of command and control helps reduce the complexity of administrating these systems. The quality of the interface and the ease of use for the automation system will impact the cost of operating the system.

The automation system should handle the details of the job execution and deployment. This will free automation planners and operators from having to master multiple skill sets on different platforms. Rather, the interface should allow the automation system users to concentrate on optimizing the manner in which the plans run. This will help minimize training costs and help provide the best value for the time the staff spends on maintaining the automation system.

Leveraging Enterprise Capabilities

Each organization needs to deal with similar problems. Systems need to communicate with one another. They need to remain secure but still provide access to the users and processes that they service. The applications are designed to interact with one another. Using these built-in capabilities can reduce cost and simplify the automation systems.

Most enterprises have the means of moving data between systems and data centers. Whether it is file shares, HTTP, FTP (or a secure variation thereof), SOAP messaging, or enterprise service buses and message queues, the systems are already in place. An automation system that can take advantage of the communication mechanisms already in place adds no more complexity. These systems are typically understood and monitored by the existing IT staff. An automation system should not add complexity to the enterprise by requiring a special communication mechanism.

However, as systems are newly integrated—through mergers, acquisition, and even consolidations—there may not be a suitable communication channel available. An automation system that can provide a channel as part of that system may provide the correct solution. It should inherently be monitored by the automation system itself, and add little additional administrative burden.

Security is always an issue. Just as people are plagued with remembering a plethora of user names and passwords to access the systems that they must use, so automation systems can be hindered or burden with the wide variety of credentials they may need to maintain. A well-designed system will provide a simple means of maintaining credentials. This capability should keep the credentials secure and compliant with credential policy.

The system should also help leverage the existing security within the enterprise. Many enterprises implement single sign-on (SSO) systems to help ease the sprawl of credentials. The automation system should leverage these systems. It should be able to manage certificates to access internal systems as a service as required. Encapsulating servers as objects to provide a single point of maintenance can also help operations easily maintain the credentials.

As systems mature in the enterprise, their ability to interact with other systems also grows. A process that may have required a custom script and external access to data may become an API and much simpler to access in a later version of the software. Many systems now provide Web service interfaces, remote function calls, XML gateways, or other similar mechanisms for inserting and extracting data. These interfaces are often more secure than custom coded systems because they are maintained by the publisher of the application.

Automation systems that can use the built-in capabilities of the applications can help reduce maintenance costs. The interfaces often remain the same even when the application is upgraded by the software publisher. When the interfaces do change, the software publishers often provide support for the migration to the new interface. An automation system that can use these interfaces will operate safely with the application and enjoy the support of the software publisher.

Job Execution

For different businesses, the triggers for running jobs vary. For some, it is merely a matter of the calendar; some jobs run every day, every Monday, the last day of the week, month or quarter, and so on. Other types of jobs are triggered by demand. The automation planners should be able to support both of these types of demand with the system that they implement.

For time schedule-based jobs, the planners need a system that can adjust to demand and the fluctuations created by the calendar. Most organizations experience seasonal changes in the workload. Some times of the year are busier than others. In addition to seasonal changes, there are times when there is more work to get accomplished. Fiscal reporting at the end of the month, quarter, semester, or year adds load to the system. The planner needs to create a system that can scale up to these increases effectively.

The business calendar can also affect job scheduling. When holidays occur, the plans may not see any of the expected workload. When organizations are multi-national and need to compensate for the different holidays celebrated in different cultures, this becomes more complex. A planner with a system that can work with calendar events will find it easier to avoid spurious error reporting and make the operational reports more accurate.

Many organizations are using event-driven architectures to respond quickly when the need arises. These plans do not execute until an event triggers the need. Events can be calls to a Web service, the receipt of a message (email, file drop, message queue, and so on). They can be triggered by network events, WMI events, and a variety of other triggers. Once received, the plan can execute. This allows the planner to create a very responsive, interactive plan.

For the automation system to work effectively, it must have a set of features that support event-driven architectures. The system must be able to respond to the events that are used as triggers. The system may be required to hold execution until all the prerequisites for the job are in place (for example, multiple input files). The system may need to ensure that the resources it needs to run are available before it executes. This can prevent the planner from setting aside resources or building “slack time” into the schedule. The system can use rules to queue plans and execute them when the capacity exists to run it.

The events may not occur on a consistent basis, so the system should have an expectation of how often the event should occur. By timing the period between executions, the system can raise alerts if a plan has waited too long to execute. This will help operations proactively monitor the plan execution without overlooking jobs that do not run on a consistent basis.

Building Durable, Resilient Job Automation Systems

Although servers may run slowly, drop offline, or fail, the enterprise still requires the information to flow and the automations plans to complete. The best designed systems will operate reliably, and may self-correct when they encounter problems.

Automation planners can build these types of features into their systems in a number of ways. For instance, some plans will have a number of servers that can execute a job. If the plan can use conditional branching, it can place the job on the server that has the most capacity at the time that the job is run.

This type of branching can be extended into a form of error correction. For example, a plan may call for a job to run on a Linux server. If the job fails to run on that server (perhaps the server is offline or too busy), the plan may be able to branch and execute the job on another candidate server. Thus, the plan executes in spite of the error. The monitoring system would also make note of the change so that the staff can correct the problem.

The servers themselves may have fault-tolerance mechanisms built-in. Database servers may be part of failover clusters, have mirrors, or have standby servers. Applications may have farms. There may be a backup data center when the primary center goes offline.

The automation system should work with the business continuity and fault-tolerance mechanisms built-in to the IT infrastructure. The more aware that the automation planner is of the failover systems, the easier it is to develop plans that will accommodate these interruptions. The more aware the automation system is of the server failover systems and provides tools to the planner to keep the plan executing even when servers fail, the less costly it becomes to build resilient automation plans.

The automation system itself is also subject to failure. If the automation system is built on a single server, it becomes a single point of failure to the enterprise automation plans. The automation system should provide an architecture that provides failover capability. A distributed architecture that allows the automation system to be run in several locations, yet managed from a single unified console, will help secure the system.

Monitoring and Auditing

Although people might not understand all that the automation system does, they may need to know some of what it does. When things go amiss, people need to know how to fix them. Some people care how the system handled important tasks and cared for sensitive data. Others need to plan ahead, so they need to understand how the system is growing and changing.

The automation system should provide insight into these arenas. The notification, monitoring, and auditing portions of the system (mentioned earlier in this chapter) will provide answers to many of these questions.

Correcting Errors

Most plans fail at the job level; the system should also help the operator identify the root cause of the failure. For instance, a job that imports data may fail because the data it received is malformed. The real culprit is not the job that failed but rather the job that provided the data.

The automation system helps in several ways. First, the system should send a notification to the operators. The sooner the operators are informed, the sooner they can take corrective action and keep the plan from falling behind schedule.

The system should provide a clear description of the error. It should provide detailed information about the error (for example, which job step has failed, why that job step failed, when the step failed, and what other jobs were executing simultaneously on the server, etc.). This provides the operator the information needed to diagnose the cause of the problem and correct it.

The challenge with troubleshooting plans is to understand the relationships and interdependencies of the constituent jobs within the plan. If a data import job fails, is it the import job itself or an error in the job that created the input file? If an event-driven architecture job has not run in 3 days, is it because the event has not fired or because the first job in the plan has an issue?

The automation system should help the operators see the plan and how each of the jobs in the plan inter-relates with the others. The system should provide information of the execution of each job step in the plan and guide the operator through the chain of causality. This will help speed troubleshooting and isolate root causes more quickly.

Validating Compliance

Five years ago, no one had heard of the Health Insurance Portability and Accountability Act (HIPAA). Now, each doctor's visit seems to require a form. People who have heard of Sarbanes-Oxley (SOX) may never have examined the ISO 2700x standards for information security. Many organizations are adopting the Information Technology Infrastructure Library (ITIL) policies for managing their enterprise, and need to prove that they are following the rules.

The need for compliance with corporate and government standards and regulations is a growing concern. Automation systems handle critical and sensitive organizational data, so they must be able to prove that this data has been handled properly. The system needs to provide the auditors with evidence that the system has completed its tasks in an appropriate manner.

The automation system itself may be governed. It will need to access critical systems to alter corporate data (particularly reporting data) and move information, so it must be carefully regulated. The security of the automation system needs to be considered. The system will store credentials that provide it access to sensitive data. It will move files with critical data. Some operations are critical to maintaining compliance. Each change to the system should be recorded. The system should be compatible with standard practices, such as ITIL, for change management.

The need for compliance reporting tends to change, so the ability of the automation system to generate custom reports that meet these changes should be readily available. Best practices for auditing require that the automation planners understand the changing needs for tracking system activity. They should ensure that their systems capture this information and can report accurately upon it in a timely manner. The easier the automation system makes collecting this data and producing the reports, the more likely the system is to remain compliant.

Capacity Planning

Monitoring also provides the basis of proactively maintaining the plans. As time goes on, the amount of time and resources a plan requires to run can change. Some jobs will take longer, while others will become shorter. Plan execution needs to be monitored on an ongoing basis to provide an accurate view of how they are operating.

Automation planners need to evaluate these changes on an ongoing basis. All the plans share enterprise resources with one another, and with other processes within the organization, so these changes can have profound effects. A periodic review of how the plans run will often reveal which servers have become overburdened and which servers have additional capacity. That can help the planners adjust the system to make optimal use of the available server time.

The automation system plays a vital role in this functionality. The system should collect a full range of relevant information. It needs start and stop times for plans and their constituent jobs, run durations. For event-driven plans, it needs to record time elapsed since the last execution and the time each execution takes.

The data should be stored in a central data repository that collects data from throughout the enterprise. The system should help provide a comprehensive set of reports that make it simple to interpret the data. Customization of reports will help suit the system to the unique requirements of the organization. As the data should be examined over time, use of On-Line Analytical Processing (OLAP) reports will help identify trends and allow proactive changes to the automation system.

Best practices for resource planning include regular reviews of the performance trends of the automation planning system. This should lead to an evaluation of the distribution of the workload within the enterprise. As the organization grows, the need for more server resources becomes inevitable. The information provided by the automation system can help determine what systems have the least capacity and help make the most efficacious purchases.

Resource Management

Each server, network channel, and resource represents an expense for the organization. IT departments need to get the most value from those resources that they can. Job automation often requires running resources at times and in a manner that does not noticeably affect the human users of the systems within the enterprise.

To get more work done efficiently, the automation planners must run the right jobs on the right servers at the right time. The challenge for the planner is that those conditions change. The planner must also focus on fluctuations in the workload and helping the server farms run as green as possible. To accomplish these tasks, they need a system that is simple to configure and modify.

Running in the Right Place at the Right Time

How and where jobs execute will often determine whether plans are completed on schedule. Automation systems that provide the flexibility to quickly and reliably change when and where a job runs can help optimize the use of corporate resources and contain costs.

Many organizations have groups of compatible servers that work together to accomplish IT functions. They may utilize application Web farms, database clusters, grids, or federations of servers. Having multiple servers running and performing the same tasks helps keep the applications responsive and eliminates single points of application failure.

The automation system should provide planners and operators the ability to use these servers efficiently. For instance, if there is a warm backup server, it may be an excellent server for the automation system to use to process data. Backup servers are often kept idle until needed. This keeps the workload off the primary servers and makes better use of the resources already in place.

Servers are sometimes taken offline and the workload transferred to another server. This may be for planned maintenance or it may be due to hardware or some other system failure. Sometimes workloads are consolidated into a single server to reduce cost. Automation systems that can move jobs from one server to another easily help to free IT to make these types of changes quickly and keeps the organization responsive.

Workloads change, and this may require the re-balancing of the servers on which jobs run. Over time, some jobs may grow and require more time and/or processing power to complete. At some point, the jobs may no longer fit within the window in which they are granted to execute. Conversely, some jobs will contract. This may leave room on some servers to take on additional workload.

The automation system can help by providing a clear picture as to the run times and resources that the plans actually consume on a day-by-day basis. Automation planners can track this data over time and use it to properly balance the plans and place jobs of the servers with the most available bandwidth.

Plans also change because the systems involved change. Software may upgrade or be replaced. New systems may provide different interfaces or mechanisms for enterprise application integration. Changes in the way a company does business may require a shift from a time-scheduled approach to a more interactive event-driven architecture.

Automation systems that help planners quickly adapt plans to meet the changing needs of the business can make these alterations cost effectively. By reducing the time and expense involved in making these changes, the system helps the IT department remain nimble and responsive to the challenges faced by the organization they service. As mentioned previously, an interface that abstracts these changes will help operators continue to keep the plans running. There should be little or no training required implementing the plan changes. Deployment can be executed from the automation console, often without touching the target servers used to execute the jobs within the plan.

Server Power Management

Many organizations suffer from server sprawl. For each new requirement the business encounters, they add more servers. Soon, the server room can no longer contain the server racks. There is not enough air conditioning and electrical power to keep everything going. The operating costs climb and the carbon footprint of the enterprise grows.

It costs money to keep servers up and running. It can cost as much to maintain an idle server as one that is handling a full load. Every server running is wearing out parts, keeping operational staff busy, and contributing to the greenhouse gases in the air.

Much of the software industry is working to help consolidate servers within the organization. Servers are better designed to handle multiple workloads. System redundancy with active/active clusters can reduce the need for warm standby servers. Virtualization allows servers to be brought online without adding new physical servers to the data center. The fewer servers the organization needs to run, the more cost effectively, and greener, the organization can be.

The automation planners can play a vital role in managing servers. They can use the automation system to make full use of the servers that are online. They can work with the existing systems monitoring systems and personnel to find the servers that have the most available capacity. They can use those servers to complete the jobs within the plan. They can also create effective plans that reduce slack time and report honestly on the resources that they require. They may find that jobs will run well on standby servers without having a major impact on the other operational systems in the organization.

The automation system can help by simplifying the process of building and modifying plans. By making it simple to deploy a new or revised plan, they can help the planners adjust the plan schedules to best use the changing conditions within the enterprise.

Periodic fluctuations in workload can also provide opportunities. It is often possible to place servers into hibernation when demand is low and bring them online when the demand increases. This can reduce power consumption and administrative workload during the slower times. Server virtualization makes this approach even more attractive. Enterprise computing capability can be scaled up or down quickly as required.

To support this, the automation planners must be able to create plans that can take advantage of the servers when they come online. Conditional branching can help support the process of determining when servers are available to take on additional workload. This helps to automate the distribution of the jobs to meet the increases in demand and the availability of resources.

Best practice is to use only the servers that the enterprise requires. This will help reduce operational and maintenance costs as well as reduce power consumption. A well-implemented automation system will help optimize the use of available resources.

Plan Configuration

The plans are a set of jobs that ultimately run on individual servers. The plans use network resources to facilitate communications. These resources, however, do not remain static. The time it takes to make changes will impact the cost and the reliability of job automation.

The reasons for change are many. Companies replace and upgrade servers and network channels. They run out of space in a server room and open new data centers to house the new servers. They outsource the servers. When the changes occur, the automation plans must adapt.

Automation planners that can reconfigure plans quickly from a single point of scheduling and deployment have a distinct advantage. If the automation system encapsulates resources as objects, the planner can easily move jobs to servers with little effort and less opportunity for error.

One of the most difficult things for IT departments to do is to merge. When companies combine through a merger or acquisition, they need to interconnect their data systems. This instantly adds a number of new servers, and even new platforms, to the enterprise. The systems within the newly formed entity must share information with one another. This usually requires the judicious use of the automation system. Many corporate system integrations stumble or fail because they cannot quickly help the systems of the two separate entities to work together.

The newly merged IT departments will also find they often have duplicate, redundant systems. This extra capacity can be consolidated to reduce the total server count, saving money on operations and maintenance and reducing the total complexity of the enterprise infrastructure. As the systems are consolidated and workloads shifted to the best servers, the automation plans will change. The automation planners need to reconfigure the automation plans to keep pace with these changes. An automation system should provide the means to deploy the changes quickly with minimal manual effort to contain the costs of the changes.

If the automation planners can quickly interconnect the information in the disparate corporate systems, the merger will occur much more smoothly. An automation system that can quickly assimilate new servers, applications, and platforms will help the planner devise a system to bridge the gap between the companies. A system that provides a single point of scheduling will help add the new systems without training the staff on how to work with the individual systems.

The automation system can enable corporations to merge more rapidly and cost effectively. Developing a job scheduling infrastructure that can adjust to these types of changes without great expense can open the door for the business to grow through mergers and acquisitions.

System Architecture

The architecture of the automation system will determine how well it can support the growth of the organization. The organization needs freedom to grow. This means that it needs to move to different platforms, migrate applications, and add and expand data centers. The automation system should scale to meet the need of the organization and help facilitate that growth. Object-oriented systems can help simplify the process of building re-usable components to configure job plans. The use of appropriate technologies can make integration to new systems simpler and faster. All these elements of the system architecture will affect the total cost of operation of the system and help determine the return on investment.

Growth and Adaptation

As an enterprise grows, its business needs change. The automation tasks required to support those changes grow with the organization. Most of this growth is incremental and organic. A business will implement an accounting system. It grows, adding applications for human resources and customer relationship management. Various departments add vertical applications to support their needs. These systems grow and change over time. Soon, an ERP system is trying to get data from a COBOL-based inventory control system that needs to provide reports to the Oracle-based data warehouse. The systems were not necessarily purchased with inter-application communication in mind, but the need for these systems to work with one another grows.

The automation planners have a combination of legacy systems that have been in place for 10 or more years trying to interface with cutting-edge systems based on service-oriented architectures. The planners need to be able to execute jobs on all the systems in the enterprise, old and new, to keep the systems sharing information. Thus, the system architecture must support the full range of systems within the enterprise.

Over time, the organization may profit from implementing new services or migrating from one platform to another. One of the major hindrances to making these types of changes is the cost to re-engineer the integration of the new systems into the existing information ecosphere. Designing and deploying a system that reduces the time and complexity of integrating new approaches and new services into the organization will reduce the risks and costs associated with these changes.

There is a tendency to think of the automation system only in terms of the servers. But the automation system integrates with all enterprise services. Security systems change with new needs and new threats to the data infrastructure. Changes in network topology also affect the automation system. Re-zoning LANs, changing WAN links, and replacing routers and firewalls all have an effect on the way that information can be communicated. The automation planners will operate most effectively if they have the tools to keep up with these changes.

The other sections of this chapter have discussed many of the changes that an organization may face. The best automation systems will help planners remain on top of the changes. It will help them track the changes that they make and facilitate testing and troubleshooting new plans and schedules. A robust system will protect the enterprise from disruption of the information flow.

Scalability

The growth on the demand for automation grows gradually with the enterprise. What often begins as a few simple jobs soon becomes a long list of jobs that run on many servers scattered across the enterprise. The best systems will be able to grow with this demand.

When developing an automation system, the planners should look ahead to likely future demands. An automation system may begin as a simple job scheduler implemented by a single department. As the usefulness of the system is proven and the need for automation grows, the demand for job automation will increase.

The best systems will scale. The system should be right-sized for the task at hand. It should be able to add capacity as required in pace with the growth of demand for job automation. A distributed architecture can often help with scalability. Running jobs locally can help reduce the demand on network bandwidth, using relatively low-cost LANs to perform the bulk of the network communications. By minimizing demand on higher-cost and lower-capacity WANs, the automation planner can use the system to help contain costs.

To facilitate the use of a distributed architecture, a single point of control is most helpful. Controlling all the portions of the automation system from a centralized view makes it easier to see the overall picture of the planning system. Automation systems that allow for distribution of the operational components can grow to handle more jobs on more servers.

Systems with thousands and tens of thousands of jobs may be difficult to control. An interface that can group views together can make the entire system easier to manage. There may be needs to provide operators with visibility of some plans but not others. An application service provider might want to publish views of some jobs to one customer, and other jobs to another customer. This may also be true for operating units within a larger organization. Scalability encompasses more than just handling more and more tasks. It should include keeping the oversight of those jobs organized and manageable.

Scalability of control may also affect the system. A rich, thick client application that provides a clear, simple-to-use interface provides a good foundation. Systems that extend to a Web-based scheduling tool can distribute control of the system without the need of installing the application on a variety of systems. The growth of mobile phones may provide the opportunity to distribute control as a mobile application that can be kept at the fingertips of the system operators wherever they may need to be.

Object-Oriented Designs

Software programmers have learned the value of encapsulating functionality into a reusable component that can be implemented wherever and as often as required. If the object needs to be changed, it can be changed in just one place and effectively update all the processes that depend on it. Developing an automation plan is akin to programming, so an object-oriented approach to plans can provide the same benefits.

A system can express each job, each server, and each plan as an object. The object contains the necessary information for that object. In the case of a job, it would contain the information on how to invoke the job. It would not, however, contain information specific to the server on which the job should run. That will allow the job to run in any server. The server objects would contain the network address and other information pertinent to the server itself. User objects contain certificates or other credentials that represent an identity on the network.

Once the objects are created, the planner can quickly assemble a job. A maintenance job may need to run on each of six servers. The job is defined only once, and a plan created to run the job on each of the servers. The job definition is reused. If the job changes, it is changed in only one place, the job object. The revised job will then run on all of the servers.

Encapsulating servers provides even more reuse. Most servers will be used to execute multiple jobs, so having a single object that defines the server saves a lot of tedious configuration. If the server is replaced or upgraded, simply reconfiguring the server object will update all the plans that use the server.

Object orientation can also simplify the development of the jobs themselves. Most common tasks have a similar structure. The automation system can be fitted with a set of standard jobs in a library. The job library contains most of the repetitive code that must be placed in that type of activity. The work of configuring a new job is simply configuring the custom portions of the job.

Event objects can be similarly encapsulated. A WMI or SNMP message can be placed in an object and used to trigger a plan. A file watcher object can be configured to monitor the files located in a given folder and trigger a plan whenever a file appears.

The benefit of an object-oriented system is typically seen in the ease at which the system can be maintained. Moving a job from one server to another is a simple matter of linking the job object to the server object. Integrating a new server into the enterprise system and using it for job automation may be as simple as defining the server object and dragging jobs onto it. A well-designed object-oriented interface can make it much simpler for automation planners to build plans and schedule them.

Architectures that Keep Plans Running on Schedule

There are a number of features in the architecture of the automation system that work together to keep plans running on schedule. Consider the impact of each of these areas on the plans and the ability to keep the automation system operating as required by the business and the SLA.

Conditional branching can allow jobs to use alternate execution paths when a job fails. This may include running the job on a different server resource or it may take the job in another direction entirely. An optimal plan can be created, but when the optimal cannot be run, an alternative execution path will ensure that the job still runs.

Notification helps ensure that operators know when a job fails. In many cases, there is only one place a job can run. If a job extracts data from a database and the database is completely offline, the job simply cannot run. It is important that operators be informed quickly when the job fails so that the operator can correct the error as soon as possible. A fast response time may enable the error to be corrected and the plan executed before others in the enterprise miss it. Systems that provide a broad spectrum of notification mechanisms will help get word to the operators quickly so that they can correct the issue.

Automation systems that provide automatic failover can suffer the loss of a server without causing a disruption in the automation workflow. If the primary automation server drops offline, a standby server can pick up the load and continue to execute the tasks. This alleviates the risk of having a single point of failure in the automation server itself. Having multiple servers can also facilitate software and hardware maintenance, allowing one server to cover while the other server is upgraded or maintained. The upgraded server can then assume the role as primary so that the covering server can be maintained. This will ensure that there is no loss of business continuity in the automation system.

In the long run, providing trend analysis of plan execution also keeps jobs running on time. By examining analytical reports that show the changes in job execution over time, automation planners can predict in advance the jobs that will have difficulty running within the constraints of the plan. The automation planner can proactively adjust the plans to compensate or begin the work of bringing in additional resources before the jobs begin to fail.

Working with the Target Servers

The best way to keep the automation system simple and minimize time training operators is to provide a common interface and scripting language that does not require the planner or operator to know the specific details, syntax, or nuances of getting a job to run on the target platform. Abstracting the manner in which plans are assembled and jobs configured will help the staff. They can build and deploy plans without directly touching the target servers. They will not avoid using certain resources because they are uncomfortable using that type of resource. The abstraction makes it much easier to schedule plans fairly and make full use of all the enterprise resources.

To make this abstraction work, the automation system needs a single execution language that will work on all the target platforms. This can be a text or XML-based language, or it may be expressed in a simple-to-use visual interface. If planners can create plans by simply dragging jobs, servers, and program flow steps into a design surface, they can concentrate their efforts on creating effective plans rather than the tedious details of syntax and semantics.

The automation system will need to use this universal language to execute jobs on the target servers. The mechanism to perform this translation can be thought of as an agent. Similar to an operating system (OS) driver, it takes a generic command and expresses that command in the specific, detailed manner in which it is understood by the target platform. Thus, copying a file is the same command in the OS, and the agent will translate it to “copy” on a Windows-based system and “cp” on a UNIX-based system. These are simple examples and the commands can be much more complex depending on the operation and the platforms involved.

Agents can be handled in a number of different manners. Some agents may be programs that must be installed on the target server. This software may be pushed to the server from the automation system server. Typically, the impact of the agent on the target is small. Nevertheless, installation of any software on the server will cause the need for some type of maintenance, the smaller the footprint the better. Systems that provide lightweight agent deployments may have less impact on the server.

Other operations may require no installed software footprint on the server. “Agentless” agents will not require maintenance on the target server and will reduce overall maintenance. Part of this will include using the Web services, database connections, or other service-oriented architectures that the system provides.

The automation system will need reliable means of connecting with all the servers in the enterprise. It is wise to implement systems with many options, even options that are not required at the time of implementation. Changes in the environment—from application re-platforming the mergers and acquisitions—can quickly make the ability to work with a different platform critical. Choosing an architecture that frees the business to make the best choices without constraint will help all of IT to optimize their operations.

The architecture of the automation system should also readily integrate with the existing enterprise. Using message queues and file transfer systems (FTP/FTPS/SFTP) that are already in place will help to reduce cost and complexity. The architecture of the automation system should be examined to see how well it can re-use existing application interfaces, communication mechanisms, and other enterprise services.

Summary

The outside world perceives an enterprise as a cohesive whole. They do not know or care that inside there are a large number of independent systems that each provide some service to that enterprise. They expect that information given to one representative of the organization will be made available to all the systems and people within the organization.

The integration of these independent systems can be daunting. It can also be critical in the success of the enterprise. The business must make clear their requirements for the automation system. A detailed SLA can help the business and the automation planning teams come to a mutual understanding of what is expected of the automation system.

Automation will require a team of professionals that can build plans, schedule and deploy the plans to run within the organization, and keep them running on schedule. Systems that make this process simple will reduce training costs and may result in more reliable automation planning and scheduling. Systems that take full advantage of the services provided by the enterprise servers often run more reliably and tend to be easier to maintain. A system that can adjust to the conditions of the enterprise can be tuned to make the best use of enterprise resources. Systems that are resilient and run even when they encounter errors will keep the system in compliance with the SLAs.

The automation system must account for its activity. Systems that can help operators rapidly identify and correct errors will reduce downtime and help the system meet the SLA. Systems that help the organization validate their compliance with corporate, industry, and government standards and regulations will minimize administrative overhead. By carefully tracking the activity of the system and providing clear, incisive reports can help planners stay ahead of problems and proactively adjust to keep the system running smoothly.

The automation system needs to be a good corporate system. It must work within the constraints of the organization. This means it must run jobs on the right servers at the right time. The automation plans must be easily altered to meet the changes that occur within the organization. They should help the enterprise reduce the number of servers they need and the power that they consume by helping make optimal use of the resources that are online and available. This means that configuration and reconfiguration of jobs should be reliable as well as simple.

The architecture of the system is the key to implementing many of these practices. The automation system should be able to “right-size,” running small and efficient when there is less demand but scaling as the needs of the organization grow. An object-oriented approach to encapsulating the elements of the schedule—servers, jobs, events, credentials, and so on—can make the system easier to maintain and faster to reconfigure. The easier it is to re-balance the schedule, the easier it will be to maintain an effective automation system. The system should have an architecture that is resilient and continues to deliver even under adverse circumstances. The abstraction of jobs into a single interface that lets the jobs run on all the different platforms in the organization will help make the system easy to use.

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>.