

Realtime  
publishers

"Leading the Conversation"

# *The Shortcut Guide<sup>™</sup> To*



# IT Workload Automation and Job Scheduling

*sponsored by*



*Mark Scott*

Chapter 3: IT Workload Automation and Job Design and Configuration .....	39
Job Modeling .....	40
Emulating Business Processes .....	40
Determining a Scheduling Paradigm .....	41
Running on the Right Resources.....	43
Target Platforms.....	44
Working with Diverse Server Platforms.....	44
Using SOA Interfaces .....	45
Batch Processing.....	46
Communication Systems .....	47
Choosing a Communication Mechanism.....	47
Dealing with Occasionally Connected Systems .....	49
Monitoring Network and Information Sharing Systems.....	50
The Power of Conditional Logic.....	51
Parameterization .....	51
Conditional Branching.....	51
Working with Business Calendars.....	52
Error Handling .....	53
Security .....	54
Secure Access to Server Resources .....	54
Securing Automation Data.....	55
Enterprise Automation Integration .....	56
Implementing Business Process Logic Throughout the Enterprise .....	56
Single Point of Scheduling.....	57
Deployment and Maintenance .....	58
Monitoring and Reporting.....	59
Summary.....	60
Download Additional eBooks from Realtime Nexus!.....	61

## 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 3: IT Workload Automation and Job Design and Configuration

The design of an IT automation job is somewhat unique. For jobs that occur on a single server or group of servers on the same operating system (OS), it is much simpler. Most OSs and applications provide systems for scripting and scheduling simple repetitive jobs. The task becomes challenging because of the variable landscape found in most enterprises.

Applications often do not run on the same types of servers. Many enterprises use a mix of mainframes; minicomputers; and UNIX, Linux, and Microsoft Windows servers. These servers often are found in data centers distributed throughout the world, separated by time zones, WAN links, firewalls, and network domains. For all their diversity, IT must help them work together as a single, seamless system that allows servers to share information and keep the entire organization current. This chapter will address many critical aspects of designing and configuring systems that can husband tasks that span the servers, applications, data centers, and networks used in enterprises:

- **Job Modeling**—The job automation processes will need to conform to business processes and adjust as business processes evolve. The process will need to compare traditional time-based scheduling with event-driven scheduling. The model must also determine the best use of appropriate resources to accomplish the work in a cost-effective manner.
- **Target Platforms**—The individual job steps of each plan are performed on a specific server or group of servers, so those servers are a key consideration in planning and configuring job automation. Applications and servers do not necessarily store or share information in the same manner, so the format of data and interfaces used to share it must be accounted for. Even the batch languages used to execute specific job tasks play a role in the maintenance and cost of the automation task sequences.
- **Communication Systems**—Sharing information between servers is at the heart of multi-system workload automation. The choice of mechanisms to move data between servers will affect the effectiveness of the plans. When systems are connected unreliably or only occasionally, the mechanisms for sharing must be designed accordingly. Monitoring communications between servers then becomes critical for troubleshooting and proactively maintaining the schedule.
- **Conditional Logic**—Plans can have individual jobs that adjust to the conditions of the moment if they have variables that can be populated and used to make decisions on how plans are processed. Implementing variables that take information from one server and pass it to another can allow conditional branching of the job steps. This can result in much more efficient plan processing. Conditional logic can also help with error correction and the synthesis of durable, reliable systems.

- **Security**—Secure organizations employ defense in depth when building networks. The IT automation system will need to deal with network security and credentials that allow it to access servers safely. Secured network channels need to be built to connect servers to one another. The credentials need to be protected and yet maintained to allow the system to operate in compliance with corporate standards.
- **Enterprise Integration Architecture**—Many enterprise applications provide their own mechanisms for sharing information. The IT automation workload system should leverage the Service-Oriented Architectures (SOAs) and application programming interfaces provided by the individual applications. Effective use of resources derived from Event Driven Architectures (EDAs) should be included. Designers need to leverage the business modeling built-in to these applications so that the logic does not need to be duplicated. A system that can provide a single point of scheduling will prove easier to monitor and maintain. The system should provide mechanisms that ease deployment and re-configuration as well as a unified reporting structure that helps operations oversee job automation functions as a whole.

## Job Modeling

Plans are meant to accomplish a business goal. There will be a process of steps that are used to convert the data that one has at hand to the information that one wants. Job modeling is converting that series of steps into a group of executable processes on the servers within the enterprise.

To create such plans, the automation planner must understand the business processes involved in producing the desired information. The planner must choose the means by which the plan will be scheduled, and how data will flow from one job to the next. The plan should make the optimum use of available server resources.

## Emulating Business Processes

The processing and movement of data follows rules and sequences. For instance, orders are placed by customers. The customer's ability to pay must be checked by credit, or the payment that accompanied the order must be validated. The items for the order must be located in inventory, then prepared for shipment. Once the order is shipped, the invoice must be prepared and forwarded to the customer. This constitutes a business process.

For many organizations, this process involves more than one computer system. The Enterprise Resource Planning (ERP) system may handle the bulk of the work, but the Customer Relationship Management (CRM) system needs to be updated with pertinent customer information. The shipment tracking system may not be integrated with the ERP system. Credit may be checked through another system.

To work in this environment, IT needs a system that can execute processes on the variety of systems located within their infrastructure. They need a system that can schedule plans to run on a timed-schedule basis or that can react to events within the organization. Above all, the system needs to take the varied tasks executed on different systems and abstract them into a single set of jobs within a plan. This becomes the heart of job automation.



For the purposes of this chapter, a single process performed on a server, such as a data extraction, reconciliation process, data transform, and so on, is considered a job. The job or jobs that work together to accomplish a business goal, such as shipping an order or generating a report, is called a plan.

Each step in the business process needs to be expressed as a job or job step that can be executed on a server or within an application. Some jobs are simple: a SQL statement that extracts data from a database and writes the results as a comma-delimited file or calls an internal process within an application. Other tasks are more complex and may require applications or custom scripts to execute several steps or convert data formats. The automation planner needs tools to help define the steps within the process as jobs that can be run on servers. He or she then needs the means to execute those jobs on a planned basis and monitor their activity. This allows the business process to be run within the IT infrastructure and accomplish the business goals of the process.

The business process models how the company does business, so systems that are flexible and easily configured and altered allow the business to remain nimble. If the organization can make a change in a business process without undue risk or extensive labor, they can rapidly take advantage of changes in the marketplace, new technologies, or new relationships with other organizations. Companies that can adapt their business processes quickly can often gain competitive advantage.

Automation systems that can support a wide range of computing platforms and formats help the automation planner by opening the doors to many opportunities. If the individual jobs can be abstracted into a system that provides a single point of scheduling, plans can be created or modified more quickly with less effort. Systems that have a proven track record for simple, cost-effective deployment of tasks in a job flow can ease the concerns of the organization as they attempt new ways of doing business and exploring new avenues of operation.

### ***Determining a Scheduling Paradigm***


Once a plan is developed, it needs to be executed on a predictable basis. The most common and simplest to comprehend is time-based scheduling. Simply, the time that the job runs is determined when the plan is designed and a timer is used to execute it on schedule.

Although commonly used, time-based scheduling is often not used well. The primary issue is the variable nature of the plans that are executed. Plans do not always follow neat schedules. Some things process every day. Other things process once a week or once a month. Some processes are run at the end of fiscal periods, such as quarterly reports and year-end reconciliations.

Beyond jobs that do not schedule regularly, the jobs themselves do not always take the same amount of time to run. Most businesses experience some form of seasonality. There are peak periods and lull periods. Holidays and holiday seasons can create brief breaks in the flow of data followed by short bursts of activity to make up for the time lost. Automation planners can include “slack time” to allow extra time to compensate for these surges. They do not enjoy full use of the available resources because if they did, some days things would not be completed.

A system that can adjust dynamically to these variations in load can help the automation planner create plans that make optimum use of resources. If resources surge predictably, the system can be designed to bring additional capacity online (such as activating additional virtual machines or powering up standby servers) and then releasing the resources once the surge has been addressed. In unpredictable situations, plans can be designed to use conditional logic that leverages other types of resources to keep the plan running. This type of flexible planning can make the system keep to its schedule automatically with little or no manual intervention.

Another option is using event-driven scheduling. Plans can be triggered into executing based on an external trigger, such as a file being stored in a monitored folder, or a Web-Based Enterprise Management (WBEM) event. A plan can subscribe to events using the WBEM model and process when the event calls for the plan. Thus, a management interface—such as the Windows Management Interface (WMI), email, RSS feed, and so on—can be used to signal when a job should be run.

 WBEM is a set of management and Internet standard technologies developed, under the guidance of the Distributed Management Task Force, to unify the management of distributed computing environments. WBEM provides the ability for the industry to deliver a well-integrated set of standards-based management tools, facilitating the exchange of data across otherwise disparate technologies and platforms.

When the trigger is received, the plan evaluates whether it is appropriate to begin. If the conditions are correct (resources are available, pre-requisites are met, and so on), the plan begins to execute. This allows the plan to check for operating constraints and dependencies before it executes. When conditions are met, the plan will run.

This type of scheduling has the advantage of using resources only when they are required and executing plans quickly upon demand. They have some other vulnerabilities, however. The monitoring system needs some level of expectation of when the plan will execute. If the plan does not execute within a defined window, operators need to be notified to investigate why the plan was not triggered. It is not known exactly when the plan will execute, so it is more difficult to ensure that resources are available to support the jobs when they run.

Automation systems that can monitor event-driven jobs and look for them to run in an expected window of operations will help the planner keep operations alert. The system that can dynamically choose the resources on which the plan's jobs execute can help to balance the resource consumption and optimize utilization.

An automation planner that has the right tools can choose the scheduling mechanism that best suits the resource allocation and Service Level Agreements (SLAs) of his or her organization. By mixing and matching scheduling paradigms, he or she can find the best way to schedule to plan while minimizing the impact on others within the organization.

### **Running on the Right Resources**

There are only so many resources in a given enterprise. As budgets are reduced, organizations are re-organized and consolidated, and workloads increase, the IT department must get more work out of every server. For an automation planner, this means they need a great deal of information:

- What capacity does each server have?
- What are the windows in which batch processes can be run?
- When are changes made to the infrastructure?
- What resource consumption will the jobs in the plan consume?
- What are the dependencies of one job upon another?
- What is the priority of the business processes that need to be accomplished?

Armed with such information, the planner can make a rough map of the resources that he or she has to work with and begin to lay out plans to accomplish the business goals with which they have been charged.

The problem is multidimensional. One consideration is that the job execution must be optimized on a given server. Often, there is little time to complete all the jobs that a server needs to accomplish. The planner must make the most efficient use of the time on the server.

The plans also need to be optimized. Some plans have many jobs running on many servers. The first jobs must run on the servers earlier in the batch cycle so that their output can be moved to the next job or plan and keep the plans operating within its designated timeframe. The planner must consider the scope of jobs for the entire enterprise, not just the list of jobs that must be executed on this server or that server.

The planner must begin with the SLAs developed for the various plans that he or she must administrate. These SLAs do not need to be complex, formal documents, but they must provide the basis on which the planner can set priorities for the various plans that he or she must schedule. Given a set of priorities and server constraints, the planner can then begin to associate the jobs with the servers on which they will execute.

The automation system can help the planner in several ways. It can make the plan easily deployed. If it is a simple matter to move the jobs from one server to another or to re-arrange the order in which jobs execute on a server, the planner will have more freedom to build and adjust the schedule. If the workload is variable, the system can help by providing options to dynamically run jobs on servers that have the most available capacity at the time the job is run. Plans that can choose resources dynamically can make the best use of the resources in the enterprise.



The planner is aided if the automation system abstracts the execution machines as objects. Object-orientation provides a single point of maintenance for elements such as security credentials used on that machine. If the machine is used in multiple plans, the machine's object simplifies maintaining those elements across all the plans.

Reporting also plays a vital role. If plan execution statistics are gathered to a central data repository, they can be analyzed. A system that helps support the building of an OnLine Analytical Processing (OLAP) data system will help reveal system trends. The trend data can be used to proactively adjust plans to keep jobs from over-running available resources. If the automation system can work in conjunction with the enterprise monitoring solutions, the performance metrics of the servers can be correlated with the plan execution statistics. This can help determine what is occurring on the servers and help planners make better decisions about when and where individual jobs should run.

## Target Platforms

It can be difficult to get applications running on the same server to share information. When server boundaries are crossed, the process of sharing information becomes much slower and more difficult. When different servers and OSs are involved, the difficulty increases an order of magnitude.

For all that, enterprises tend to have a variety of applications running on different servers and different platforms. Much of the challenge of job automation involves getting these platforms to work together to achieve a common business goal.

Each application and OS will have some mechanism for job automation. Each one will be unique to that platform, yet the automation planner must use them all to accomplish his or her goals. The format in which the data is stored often needs to be converted before the next job in the plan can use it. Even the scripting languages themselves will vary by platform.

### *Working with Diverse Server Platforms*

Most organizations grow organically. They start small and expand over time. They start with a single departmental server, and add servers to support new applications, such as email and accounting. Maybe they purchased a minicomputer or mainframe years ago and still use it as a core application in their enterprise. Perhaps they purchased or merged with another company that used nothing but Microsoft servers, and now their corporate applications are a mix of Microsoft, UNIX, Linux, and OpenVMS servers, ranging from 20 years to 20 days old.

Although each server has a distinct purpose within the organization, the applications they host and the data they support must work together to provide a cohesive picture of the organization. Thus, the applications and OSs must be coaxed into sharing that information.

Job automation systems must find the means of getting those applications and OSs to execute jobs on a scheduled basis. This proposition is interesting because each application and each OS has its own distinct internal means for scheduling and executing jobs. If a planner were to schedule across all the platforms without some type of tool, he or she would need to know how to schedule jobs in OpenVMS, UNIX, Linux, and Windows. The planner would need to be able to go to each server and adjust those schedules on a regular basis to keep operations in sync with one another. The task would be overwhelming.

Job automation systems help by abstracting the process into a single console. The scheduling system can launch jobs on the individual target servers. Where required, agents are installed on each of the target systems. The planner needs only to configure the job to run and determine how it is scheduled (through time or events). The automation system handles the details of triggering the job and monitoring its operation.

The planner can then step back and view the sequence of jobs that are required to create a plan that fulfills a specific goal. The individual jobs that comprise the plan to fulfill the goal need to be laid out in sequence. Some jobs steps can be run in parallel, others need to run sequentially. The planner needs to map a plan that pulls these individual steps together and gets each step to run on an appropriate server.

These jobs often need to be run on a variety of servers. Finances may run on UNIX servers operating Oracle Financials. The shipping system uses a Windows server to track which orders shipped and how. A different system, running in Linux, collects hours worked data. It all gets stored on a database server, and a different server is used to process the OLAP cubes. Reports are processed on yet another server, and so forth.

For an automation planner, the jobs run on individual servers should be abstracted and made simple to run. If the planner can place the job in the proper place in the queue and execute it reliably, they have the freedom to design plans that are flexible and effective. The automation system can help abstract the jobs. The system can provide agents that can handle the esoteric details of executing a job on a target platform. This frees the staff from knowing the myriad different means of scheduling jobs on different platforms and frees the planner to develop the most efficient plans. If the servers, queues, and other system components are abstracted as self-contained objects, it simplifies the task of building or revising the plan, and makes the plan easier to maintain.

### ***Using SOA Interfaces***

Enterprise applications, such as ERP systems, accounting systems, inventory control systems, CRM systems, and so on, often need to share information with other systems, often on other platforms. This has given rise to SOAs. Through this architecture, systems provide platform-agnostic interfaces, such as web services, to allow external systems to execute procedures and access data.

For automation planners who need to help systems work together, these interfaces provide the simplest and more reliable means of interfacing with enterprise applications. The interfaces are maintained by the application publishers, so all the automation planner need do is use the interface in the prescribed manner to perform any available operation.

These interfaces are not without their challenges. Such interfaces may be manifest in different forms, not limited only to web services. Whether it is iDocs, EDIFACT, or some custom format, the document schema must be stored and dealt with. The variety of formats must be handled and the inherent contracts with such documents must be honored.

The interfaces also require security. No uniform system of exchanging security tokens or credentials exists (although some single sign-on—SSO—systems may provide a semblance of this functionality). The automation planner must account for the variety of different means of accessing these interfaces.

The job automation system and its capabilities play a major role in leveraging the SOA interfaces available for use. If the system is designed and implemented to use such interfaces, it greatly simplifies the automation planner's task in leveraging them. If the system can manage the security credentials or tokens in a secure but maintainable manner, the resulting plans will be easier to keep running.

Although most applications maintain compatibility of such interfaces from one release to the next, they will inevitably change. New interfaces will be added and old ones obsoleted. The changes will slow updating improved or patched versions of applications that the entire organization depends upon.

Some SOA systems—such as Microsoft BizTalk Server, BEA Aqualogic, IBM Rational Asset Manager, and LogicLibrary Logidex—may handle a significant portion of the process, working with multiple systems to complete a plan. These integration systems cannot always complete all the steps in the plan. They also may require a different means of scheduling and reporting than other automation tasks within the enterprise. If these systems can be overseen by the automation planning system, a single point of scheduling and monitoring can be maintained.

The automation system should provide flexible means for adjusting plans when these interfaces are altered. Automation systems can deploy revised plans quickly and reliably and make such upgrades easier to test and faster to implement. And, on those occasions when upgrades or changes are implemented in an unanticipated manner, an automation system that strongly monitors and clearly reports the operation of the plan will help IT quickly identify the issue. The automation planner can use this information to remediate the errors and put the plan back into production.

### ***Batch Processing***

Virtually every OS and most applications have some type of batch processing language that can be used to execute tasks on the platforms and products which they serve. Each language has its own unique manner of executing, processing, and reporting errors.

The range of languages would prove a challenge to any single programmer. Some will run in JCL, others in REXX, some in COBOL. UNIX and Linux shell scripts, DOS batch files, and VB Script all play a role. Databases offer automation from command-line shells, as do many other applications.

Automation planners need a consolidated mechanism that can perform the same functions and/or launch these scripts on demand. This will allow the plan to execute jobs across platforms and set control of the process in a single console.

The automation system should allow for the launching of the scripts. Since the threads on which the batch processes are launched will have an inherent identity with certain inherent rights, the scripts must be launched with the appropriate identity.

The automation must also be able to monitor the execution of the script and record its status to a central repository. This repository is the source for validating the operation of the automation system. It provides an analytical source for projecting the workload executed when jobs are run. It also provides a source for auditing system activity and providing evidence frequently used in assuring system compliance with corporate and regulatory standards.

Most scripting languages provide little or nothing in terms of structured logging or error handling. The automation planner needs a system that can monitor a variety of sources for such information. It may take the form of text files created by the system, database entries, system event logs, SNMP messages or other formats. The automation system needs to monitor these sources and alert operators if jobs fail. It must also keep record of the errors encountered to provide the means for operations to perform forensics on errors and plan corrective action.

The automation planner is either restricted or released by the capabilities of the automation system to work with the batch scripting mechanisms used within the organization. The inability of the automation system to work with any particular means of executing batch processes will either require the process to run outside of the jurisdiction of the system or cause the automation planner to avoid using that batch mechanism altogether. Systems that provide a wide range of support for batch processing mechanisms will provide the greatest flexibility for the organization. The system should be batch and script language independent. It should support all the batch languages in use within the enterprise to ensure compatibility with all the systems on which it must schedule jobs.

## Communication Systems

Since the first company ordered its second computer, passing information between systems has been vital and difficult. As networks became the norm and enterprises needed to connect computers in remote locations, a variety of mechanisms were developed to share information.

The challenge for the automation planner is to choose the best of the variety of mechanisms for moving data between systems in the enterprise. The planner must deal with the performance constraints and reliability of the communication systems employed. He or she must also be able to monitor the system and ensure that plans proceed from one job to the next as designed.

### *Choosing a Communication Mechanism*

There are a wide variety of mechanisms to choose from when sharing data:

- File shares
- FTP/SFTP/FTPS
- Job queues
- Web services
- Database records
- Inter-Process Communications (found in a wide variety of types)
- Email

For many tasks, there are multiple means of sharing that could be used. For the automation planner, there must be criteria for choosing the appropriate one within a given plan. There are several criteria that can be used to help make this evaluation:

- **Which mechanism is the easiest to implement?** The best choice is one that is implemented, supported, and understood by the enterprise. New and alternative communication channels may offer unique advantages, but these benefits must be carefully weighed against the increase in implementation and support costs.
- **What mechanism is easiest to monitor?** If a system uses a file drop but nothing monitors whether the files are being retrieved, the plan has a built-in recipe for disaster. The communications mechanism should be easy to monitor and ensure that data is flowing. It should also be easy to correct when errors are detected.
- **What mechanism is sufficiently secure?** Many automation jobs move sensitive data. The channel must be secure, even at the cost of performance or additional maintenance. Virtual Private Networks (VPNs), secure File Transfer Protocol (FTP), Secure Socket Layer (SSL), encrypted files, and file system folders all play a role in designing a communication channel that can protect the data that it moves.
- **Which mechanism is most reliable?** Not all methods of connectivity are equally dependable. If synchronous connectivity is desirable for extracting data from a JDBC data source, dial-up modems may not be the best choice. Some WAN connections can be very temperamental. The planner must understand the nature of the connections and work with their limitations
- **Which mechanism is most economical?** Systems that use wide area networks often have limited bandwidth. The cost of connectivity can become high. The art is choosing the mechanism that is the most cost effective, balancing performance against cost.

The automation planner must consider all these questions when choosing the means for conducting data between systems. The balance of performance, security, reliability, and cost is not an easy one to strike. The planner may need to experiment to discover the channel that best answers the need.

The automation system can open possibilities to the planner. The wider range of communication options that the system supports, the more choices that the planner has at his or her disposal. If the system is able to monitor using mechanisms such as SNMP, the system can better track the performance of the communication system itself.

Communication systems are no more static than the servers. Network paths change over time. VPN, WAN, and VAN links have security requirements and may change as well. A flexible system that can alert operations of unanticipated changes and allow data transfer patterns to be changed quickly and easily will help keep the plans operating with minimal disruption.

### ***Dealing with Occasionally Connected Systems***

When systems are all connected through reliable LAN or WAN connections, it is much simpler to keep data flowing on track. But for many enterprises, there are systems that do not remain connected. There are a number of reasons that the servers may not have continuous communications with one another. If an organization has a large number of branch offices, it may be impractical to network all of them. They may use VPNs or dial-up solutions. Some have small bandwidth links, and so, even though the site remains connected, the use of the connectivity must be carefully rationed. Some remote locations may not be able to obtain reliable service, so connections must take advantage of connectivity when it is available and hold information when it is not. Some data may be located in mobile sources and require transfers when the connection can be made.

When systems are occasionally connected, the planner must design the job to suit the requirement. This requires an understanding of how the systems are connected. If the connections are regularly scheduled, the plan can build around that schedule and include connection as one of the jobs within the plan. That simplifies modeling and helps make the plan easier to monitor.

Job performance becomes an important issue to consider in these types of connections. Occasional connections are often lower bandwidth, so the amount of data has a strong direct impact on the performance of the job. The window of connection for the job is often specific and defined. If the job begins to take longer than anticipated, the window may not suffice. Steps such as compressing data or preprocessing (to summarize and reduce the amount of data) may be required to help the job fit in the desired window and bandwidth of the connection.

If the connections operate on demand, the plan should be designed to use on-demand interfaces. The monitoring should use techniques suited to on-demand scheduling. There must be some expectation of the time or interval of the connections so that they can be monitored and operators alerted if the connection waits for too long a period.

If the connection is unreliable, the plan must be designed to compensate. For instance, a low-cost connection to a remote site may be preferred but not always operational. The plan should allow the jobs to use the low-cost connection. If, however, the low-cost connection is not available, the plan should use conditional logic to choose a higher-cost connection option.

To allow for these variations in plans, the automation system must support the options that the planner needs to implement the most effective solution. The system should be able to monitor communication channels. This may require the ability to read SNMP messages or integration with the enterprise solution monitoring system. If the connections are not driven by a structured schedule, the system should track how often jobs are run. If a job waits past a set threshold, the system can let operators know that some corrective action should be taken.

If the window is limited and the job grows, it must be carefully monitored to ensure that the job can be completed within the window (or the connection) is lost. The monitoring system should help planners identify jobs that cannot pass their data through. The planner can then make adjustments to provide the additional resources that these jobs or plans require. A central repository of monitoring data can be used to analyze the trend in job growth and help the staff take proactive steps in preventing the risk from becoming a realized problem.

### **Monitoring Network and Information Sharing Systems**

Data is moving through the network, so monitoring the network is an important part of every automation plan. Some jobs are quite simple: the data moves between servers located on the same LAN within the same data center. For this type of enterprise, network monitoring is not very essential.

But while IT has moved predominately to TCP/IP flowing on Ethernet networks, that is certainly not the only way that data is moved. When dissimilar networks interconnect, the connections should be monitored for disruptions. A disconnect in the flow of data can be difficult to diagnose because the jobs involved will not fail—only the movement of data fails.

But for many organizations, automation will encompass moving data between multiple locations. This will involve the use of some type of WAN. Awareness of the condition of the WAN can allow the system to intelligently choose other routes. Also, the automation planner needs to know when bandwidth is limited and when it is more available. This knowledge will help adjust the execution of the plans to optimize the use of network resources.

Some systems are more difficult to track. For instance, message queue systems work more like applications than transport systems. Messages can enter the queue and fail to exit. The reason for the failure will be found in the message queue logs but not generated as a specific job failure.

The automation planner needs a thorough understanding of the sub-systems used to move data between jobs within the plan. He or she needs to ensure that the system can keep atop the movement of data and can detect when and where data movement or messages get stuck. He or she might also want to add conditional logic that adjusts the flow of the plan based on the conditions found within the network and transport mechanisms themselves.

The automation system plays a vital support role in building these plans. The system needs to correlate the activity of the job plan to network conditions. This can be done through a direct ability to monitor network system and/or integration with enterprise monitoring systems. The correlation can be used to help troubleshoot problems that occur in the plan flow, even if no job specifically generates an error.

Conditional logic based on network performance can also be used. It can choose lower-cost network paths when they are available or choose higher-cost paths when the lower-cost path cannot be used. Plans can be designed that choose whether to process jobs on local or remote servers, accounting for both server and network availability. This will help optimize resource utilization throughout the enterprise as a whole.

Network outages cause plans to fail to execute. The plan should be designed to detect the health of the network and alert operations staff quickly if the network fails and the plan cannot be executed. This will help fix the problem without causing undo delay in the execution of the plan.

## The Power of Conditional Logic

The conditions under which plans execute change frequently. The changes may be due to changes in the data itself or to changing conditions within the environment. If the automation can define these changing conditions and allow the plans to automatically adapt, the system can make better use of the internal resources and run more effectively.

### *Parameterization*

Plans can be parameterized. This allows a single plan to be executed in several ways based on the parameter. For instance, the same plan may need to read three different source files. Once read, the data is processed in an identical manner. By creating a single plan with a parameter that indicates which schema to use on the source file, a single plan can serve all three data sources. The business logic and flow of the plan is contained in a single place. There is less code to maintain and all the business logic resides in only one place. This can help improve reliability and reduce maintenance costs.

### *Conditional Branching*

The workflow of a job may also need to change based on the exit codes of a job within the plan. Conditional branching allows a job to process and pass information back to the scheduling system that will allow it to determine the next step in the plan.

The key to such changes are dynamic variables. The variables can be interpreted by the plan as it executes and cause changes to the order or location in which individual jobs execute. This allows the planner to move jobs, skip jobs, add jobs, or make other changes as required to keep the process running in an efficient manner.

The changes should be flexible enough to report normally so that the monitoring and auditing of the system runs normally. That way, the system can still determine what constitutes an anticipated change and what constitutes an error and appropriately inform the operations staff.


The automation system is vital to this type of automation. The system needs to handle the flow of the plan from one job to another independent of the jobs themselves. It must support the creation and population of the variables from the correct source. For instance, if a job produces no output, and this is an acceptable outcome, it should be able to populate a dynamic variable that will terminate the plan without an error. If servers or other resources are available, a dynamic variable should be populated to allow the system to use alternative resources to execute the job without throwing the schedule out of kilter.

Although dynamic variables and alternative plans are not absolutely required for the creation of an automation system, they add a level of control and adaptability that will free the automation planners to take full advantage of the resources available within the enterprise. A creative planner can use this capacity to devise systems that are much more durable, reliable, and efficient than systems that are confined to a single path of execution.



### **Working with Business Calendars**

Work does not flow evenly within an organization. Almost all organizations experience some type of seasonality. All companies deal with local holidays that cause a temporary cessation of work. Many tasks are cyclic based on the fiscal calendar with weekly, monthly, quarterly, and year-end reconciliations. Plant shutdowns and re-tooling periods can also conspire to make the batch schedule erratic. If not properly planned, it can make the system very difficult to manage and monitor.

 In this context, a calendar denotes a time filter. The calendar allows a schedule to be set for a common interval (for example, every Monday) and then note exceptions to the schedule (for example, New Year's Day). The calendar helps manage the exceptions and variations to the normal rules.

All these events will cause fluctuations in the plans that are run and the schedules that are kept. For the automation system to fulfill its task, the system must be able to distinguish between planned disruptions in the schedule and legitimate problems that should trigger alerts.

An automation planner should be able to build a complete calendar of events into the system that he or she devises. The calendar should help the operations staff and the system understand variations in operation, processing time, and resource requirements in context of the variable needs of the schedule. The more automated this process, the less it will cost to execute.

Days will occur when there is no business activity. They may be holidays, plant shutdowns, or other events that happen on a scheduled but irregular basis. If jobs are scheduled to occur simply “daily,” on these days, they will fail. The system may correctly notify the staff of the failures. Since it is a holiday, everyone will understand and ignore the failures. The system should not generate errors when something as common as a holiday occurs.

Some jobs are business-oriented and need to adhere to the business calendar. Other jobs run day in and day out, holiday or not. The planner should be able to configure the difference within the system so that it distinguishes each type of plan correctly and operates it appropriately.

If the automation planner can set the system to recognize the holiday and indicate which plans will and which plans will not run, the operations staff will be presented only with legitimate alerts. It will make the system auditing and reporting more accurate as well.

The automation planner can also take advantage of these slack times to run automation plans that help maintain or purge systems. This can make the overall system more efficient while taking advantage of the available resources. If these maintenance plans can be scheduled to run during holidays, it will minimize the labor of setting them to run manually.

The other end of the spectrum is jobs that run periodically that add load to the system. Reconciliations, seasonal increases, and other sources can add to this load. The added load can affect the performance of the scheduled plans and add plans that run on a sparse schedule.

The automation planner may need to adjust the plans to accommodate these changes. Resources may be held offline when not required and then powered up when the added load is required. Scheduled tasks may need to run on alternative resources or run on modified schedules to allow for the added workload.

Providing a system that knows when these times are expected and that can make these adjustments keep the automation tasks on target. By designing a system that can change behavior automatically to deal with a variable workload, the planner can devise a system that always requires and uses only the minimal set of resources. This can help contain costs.

If the system can accommodate this type of design, it frees the planner to make the best use of enterprise resources. It can also accurately monitor the system. By correlating the business calendar with performance monitoring, it makes analysis of the system more accurate and easier to understand. It also can minimize false alerts.

### **Error Handling**

For most programming systems, error handling presents one of the greatest challenges. There are basically two types of errors—those errors which a planner can foresee and those errors which the planner cannot foresee. Expected errors can be encountered for a number of reasons. For instance, a job may run on a regularly scheduled basis. It may pick up a file from an outside source or another step in the plan. If the file is not in the expected location, the job cannot be completed. This is an acceptable error.

The automation planner can create a plan that compensates for expected errors. In the previous example, the job may wait for an interval and try again to collect the file. The system may have a specified number of retries before the plan is considered to have failed. Failure can occur from busy resources, failed connections, and other hindrances. A plan that can compensate for the error and continue to run will keep the automation system operating.

A planner with the right tools can help the system correct itself when it encounters expected errors. If the system self-corrects, it reduces labor and helps maintain the integrity of the automation plan. The proper reporting can help uncover the source of the errors and make appropriate adjustments.

The planner will also encounter unplanned errors. Sometimes the system on which a job is scheduled to run fails. When the system fails, it cannot report the error. It often can provide little in terms of useful forensics. A system that monitors the progress of jobs from systems other than those on which the job runs can more easily detect the error. They can provide information about the job execution that helps the operations team determine and correct the source of the error. The monitoring system can notify operations staff even when the executing system was unable to raise an error or report the problem. The planner can use this capability to ensure that the plans are executing and determine whether the plan needs to be altered to make it run more reliably.

Providing an automation system that runs independent of the plan provides an independent witness to the execution of the jobs themselves. It provides a consistent reporting and notification source that is not affected by the failure of the systems on which the individual jobs run. A system that can compensate for job failures can allow the automation planner to create more robust, durable systems within the enterprise.

## Security

There are many considerations when addressing security within an automation system. Individual jobs run on servers and require a security context. The automation system must be able to run in an appropriate security context within those servers. Beyond that, the data handled by automation often contains sensitive information. The data itself must be secured within the enterprise while it is moved from location to location.

### **Secure Access to Server Resources**

When a job runs, it requires an appropriate security context. This may be no more complex than providing credentials when invoking the job. It may involve storing and retrieving security tokens, such as user certificates or tokens. It may require identity delegation or impersonation.

One of the challenges within a heterogeneous environment is that different platforms require different forms of authentication. Some forms of authentication are required at the OS level, such as Windows Security Identifiers. Some require authentication against directory stores, such as Lightweight Directory Authentication Protocol (LDAP) or Microsoft Active Directory (AD). These forms of authentication will require the job to run within the authority of the account that the security mechanism identifies. Other systems and many applications may have their own means of authentication. Some will depend on a string representing a user name and a password. Others will require more sophisticated mechanisms, such as a user certificate.

The credentials used to run the jobs often provide access to sensitive information. They will provide rights that the staff who operate the jobs do not and should not have. The rights of these accounts often provide elevated privileges on the servers on which they operate, so they must be kept secure.

Most enterprises institute policies that require user access credentials to be changed on a periodic basis. This helps to protect against the times when credentials have been compromised. It also means that the credentials issued to the automation system will need to be updated on a regular basis.

The automation planner needs to be cognizant of the security requirements of all the systems that the jobs in the plan touch. The planners want to ensure that the job has sufficient rights to execute the requirements of the job. That means that the automation system will need to store and present the credentials to every system on which jobs will be executed. Systems that abstract the underlying components and store the security credentials in a single location keep the system more easily maintained.

The proper automation system should be designed with the full range of security requirements designed into the product. It should provide secure storage of the security tokens, certificates, Kerberos tickets, user credentials, or the like. It needs to provide a mechanism that makes these credentials easy to modify when required. It also needs to store the credentials securely so that miscreants cannot easily use the automation system to gain access to sensitive corporate systems or data.

Security and access to data are often the topics of corporate or legislative regulation. The location and use of the credentials may require careful auditing and tracking. The automation system should provide clear records of the access made to the credentials themselves and an audit trail of how the system used those credentials to perform its designated tasks.

Some of the most common failures in automation systems are centered on the inability to access resources. The system should report security errors in a clear, concise manner. This will help the planner during the initial testing and deployment of the system. It will also assist the operations team later when jobs cannot run because security requirements change.

### **Securing Automation Data**

The automation system jobs will often extract data from applications to share with other applications. The data needs to be exported and imported in an interoperable way, so this may be the time when it becomes most vulnerable to theft. The automation planner needs to plan to keep the data secure while it is moved within the plan.

For example, the orders that a company receives each day may be part of a data extract that is placed in the data warehouse. To perform the extract, the data may be placed in a comma separated value file and placed on a file share. Another job may later read that file and process it for the data warehouse.

The file would contain sensitive information. The data may include personal identifiable information (PII) that is regulated by law and the industry in which organization operates. The data may contain information that would be very damaging if it were given to the wrong individuals.

Such files need to be protected. Encryption of data and the use of secure file transfer systems can help shield the information. Thus, the planner needs to use a wide variety of security mechanisms. Many OSs provide file and folder encryption. Data can be moved by secure transport systems, such as SSL transports, L2TP encrypted channels, VPNS, or secure FTP. Although all these systems are commonly available, they vary from one operating platform to another. The automation planner must be able to work with the mechanisms at his or her disposal to ensure at all times that the data is adequately guarded.

The automation system can put a wealth of tools at the disposal of the planner. It should be able to leverage the tools available from the applications and OSs to keep the data secure. It should provide the means of keeping exposed data encrypted, whether directly or by using the mechanisms provided by the operating platforms.

The channels used to move data can be easy to overlook but often represent the greatest risk. When data is placed on a LAN, every network card on that LAN segment can read the data. When public WAN links are used, the data can be passed outside the secure borders of one location in the organization into an exposed and easily violated cloud of unprotected network traffic.

The automation planner must use the system to move the data securely. The use of encrypted channels, such as SSL connections, VPN links, and other secure network connections can encrypt network traffic while the packets are vulnerable. The system must be able to support the use of these secure modes of communication. It must also monitor to ensure that the channel is operating correctly and the data remains protected throughout the process.

## Enterprise Automation Integration

The automation system works within the applications that are installed throughout the entire enterprise. The system must touch and build interactions with systems of all types, wherever they are located.

Many of these systems encapsulate a portion of the business logic that the plans represent. The automation planner needs to leverage that portion of the business logic with minimal duplication—then extend it to other systems. A system that provides a single point of scheduling can help coordinate the system across the entire organization. These systems may be located far and wide and the landscape of enterprise data systems is seldom constant, so a mechanism for deploying and maintaining plans will help make operations much simpler. The automation planner must also provide a mechanism for monitoring the system and reporting on the results.

### *Implementing Business Process Logic Throughout the Enterprise*

Most systems model business processes within the scope of their operations. An ERP system will provide the steps of processing an order and moving that order through the necessary stages to complete the order, deliver it to the customer, and collect the invoice. The system contains the means of executing the individual steps in that process and integrating functions of accounting, inventory control, production management, human resource management, and customer relationship management. So why must the ERP system integrate with other systems?

Most organizations use a collection of systems to perform these functions. They may have legacy systems that perform some of the functions and find that it is not economical to move the business logic into another system. They may have merged and inherited another set of systems that work fine with their sister organization and cannot justify the time, effort, and expense of standardizing on a single system. Sometimes the peripheral systems provide functionality simply not found in the master system. In some instances, it may simply be a matter of hardware costs and software licensing.

For all these reasons and more, the enterprise is often a cornucopia of different systems, each of which perform some of the operations required by the corporation. Automation needs to link these systems to form a cohesive whole. The automation planner needs to leverage the business process and logic found in each of these disparate systems and knit the whole into a comprehensive gestalt that represents the business processes of the organization as a whole.

It can be difficult to allow a system to perform its function and still help it to share that information with other enterprise systems. Most enterprise systems provide some mechanism for sharing information with other systems, but the mechanisms vary from system to system. The automation planner must use these mechanisms to allow each system to do its part and knit the whole together to ensure that all the steps in the process are completed.

This will require having an automation system that communicates with each of the disparate systems. A wide variety of communication mechanisms are required. Some will use web services; others will provide simple file exports. Some require connection to database packages and procedures. Other systems provide little in communication mechanisms and require the automation system to do most of the work in collecting and organizing data for other systems to work. A well-crafted automation system will provide the planner with tools that enable him or her to fully leverage the business logic encapsulated in the source system and move the output of that process to another system. It will provide the means to move it efficiently and direct it to the optimal resources to consume the information. The resulting system should abstract the complexities of dealing with the underlying application into a simple console that executes the job, monitors its run status, and moves to the next job in the plan.

The less business logic handled outside the source systems, the simpler and less costly the automation system is to operate and maintain. By using the business logic built-in to the source systems and communicating with them in the manner in which they are designed to communicate, the automation planner is freed to build the plans and balance the use of resources.

### ***Single Point of Scheduling***

The enterprise is a single organism where every individual component has some effect on every other component. In plan automation, the systems have dependencies on one another. Although some tasks can be run in parallel, many are serial. A job cannot process an input until that input has been created. Thus, the time the data extract on the previous job runs will determine how soon the information can be processed into the next system.

Although an automation planner can spend ample time preparing the jobs in an individual plan, the real art of the position is getting the jobs to work in harmony so that all the plans are executed on time. The ordering of individual jobs on a given server requires the ability to prioritize the jobs and see how the execution of each job affects the other jobs within all the plans that must be run.

There is no simple way in which to perform this task. One of the best analogies for this is found in filling a jar. A person fills a jar with large rocks until it appears full. He then pours pebbles into the jar, filling the spaces between the rocks until the jar appears full. He then pours sand into the jar, filling the space between the pebbles until the jar appears full. He then pours water into the jar. The moral of the story is to place the large rocks into the jar first.

The planner needs to identify the plans with the longest and most complex job steps and schedule them first. He or she must then work around this schedule to add less complex jobs into the gaps in the schedule, working first the jobs with the most dependencies to the jobs with the fewest. By carefully layering the jobs from various plans into the server, he or she can get the most effective schedule for that server.

To accomplish this, the planner must have the full complement of plans available to him or her. He or she must be cognizant of two things. At some point, the server will indeed reach capacity and be unable to service any additional jobs. And, when that happens, some plans are more important than others.

A single point of scheduling is invaluable when building these types of schedules. It should provide a view that shows where the individual jobs for each plan are executed. It should show how long the entire plan will take to run. This information will help the planner “shake” the jar and get the jobs to fit in the spaces between other jobs.

A single point of scheduling can also help identify where other enterprise resources may be available. There is a tendency to assume once a job is allocated to a server that there is no other place where it can be executed. Systems that make changes to scheduling and deployment of jobs difficult help foster this misconception. A system that can help the planner identify other locations where the job can be run will help the planner balance the load across the enterprise in the most advantageous location.

Most enterprises will undergo changes. When servers are moved, consolidated, added, or re-tasked, the jobs for the plans need to be re-scheduled. A single point of scheduling will help the automation planner see the interrelationship of all the plans. Seeing how the jobs affect the overall schedule will simplify the process of re-scheduling and reduce the likelihood of errors or disruptions of one plan by re-arranging the jobs in another plan.

## Deployment and Maintenance

Designing plans helps the organization only if those plans are properly installed and configured. Much of the labor cost of the automation system can be categorized in setting up jobs on servers and making changes to the system over time. If the system can help deploy changes reliably and with minimal effort, it will help contain costs while freeing the automation planner to continue to refine and optimize the automation plans.

The variety of platforms presents a significant challenge when deploying plans. Each system will have its own means of executing jobs. The planner needs to be able to place the job on the system, control when it will run and provide any required security credentials. The less manual effort involved, the faster, more reliable, and more cost effective the deployment will be.

To help streamline the process, the planner wants a system that allows the plan to be created from a uniform console. The jobs should be distributed from the console to the target servers automatically. The job should also contain all the elements that it needs to execute on the target server. The automation system will need the ability to configure and monitor the job remotely.

Most automation systems provide an agent structure. The job can be created with a uniform interface. The agent translates the canonical information into the specific details required to execute that job on the target server. The agents help abstract the technical details of configuring the job on the target server, so the planner requires less arcane knowledge and less support from specialists who maintain expertise on the target platform. This will help simplify the deployment of the plan.

Over time, the plans will need to change. The changes are driven by changes in the business process, changes in data volume, modification of existing systems and infrastructure, and a variety of similar events. As the system changes, the schedule will no longer serve their design purpose.

The automation planners should regularly monitor the performance of the automation plans. Using the data stored in a central repository, they can use analytical reports to determine the appropriate changes to the plan to remain proactive. Some plans will run in less time than anticipated. This will allow re-arrangement of the other plans and may allow for better utilization of resources or an easing of tight timeframes. Other plans will take longer over time. By seeing the trend in growth, the planner can anticipate problems and re-direct resources before they begin to generate errors. This will keep the enterprise systems running with less disruption.

The automation system needs to provide key features to support the planner in these efforts. The system must have a wide range of agents that allow deployment to all the various servers and applications used within the enterprise. Planners should look ahead—not just to the immediate need—by finding a system that will support platforms likely to be added to the enterprise through mergers, acquisitions, or application upgrades or migration. The more platforms that the automation system can cover, the more nimble the organization can be when building their enterprise information infrastructure.

The system must also make the deployment and re-configuration of plans very dependable and quick. If new and revised plans can be deployed with minimal effort, the planners can remain nimble in fine tuning the use of the enterprise resources.

### ***Monitoring and Reporting***

Monitoring is vital to the automation system. The organization must have assurance that systems are executing as designed and that the automation system is fulfilling its designated role. The reporting repository can help provide the basis for system analytics and satisfy regulatory requirements.

The first need is for job monitoring and alerts. When jobs fail, the operations staff needs to be notified quickly. They need to be able to access reports that show the flow of the plan and point of failure. The reporting should help isolate the fault and help the staff restore the execution of the plan with little delay. The planner needs tools to collect this information and configure clear, diagnostic reports. Integration with enterprise server monitoring systems can also help if the errors occur at a server or communications system level.

The planner needs a complete set of reports that consolidate the view of automation execution throughout the enterprise. The planner needs to see the effect that one job has on the schedule as a whole and the other jobs that are run on that server. Reporting that integrates operational monitoring data to the automation plan reporting will help the planner better see the overall picture of how the automation plans are running. This will help the planner make better decisions on plan adjustments.



Some plans have regulatory components. When the organization needs to prove that data has been kept secure and that certain IT tasks have been completed on schedule, reports will be required to satisfy the auditing requirements. The planner needs to provide these audit reports as part of the fulfillment of the plan requirements.

An automation system with robust reporting capabilities will help the automation planner fulfill his or her responsibilities. The system should provide a central data repository from which a wide range of reports can be derived. The ability to customize reports to meet specific needs is useful. The system should help the planner develop analytics that show the trends in the plans. This can be used for capacity planning and schedule refinement.

## Summary

The purpose of any IT system is to automate portions of the business process conducted by an organization. The differences in the way that businesses conduct their affairs constitute their individual competitive edge, and thus must be conserved. When designing an IT automation system, the planner needs a system that can closely model the business processes of the organization. That will mean using a variety of approaches to job scheduling, from time-based to event-driven. The system must scale to meet the demands placed upon it by the business process requirements.

There are so many ways to do the same thing, so there are many computer platforms and applications that do them differently. An organization must choose the components that best suit their way of doing business. That means that the automation system must support the diversity of platforms embraced by the enterprise.

Communications for many organizations span countries, continents, time zones, and cultures. There are a variety of communication paths and mechanisms to support this geographic dispersion. The automation system needs to deal with the different means of connecting one location to another.

There are many plans that will run differently one day than the next. A change in the data or conditions in enterprise resources may prompt an alternative plan execution. Changes in the business calendar may also direct changes. Sometime errors call for compensating plans. A planner that can use these variables as part of the plans themselves can build a more robust, self-directed system that runs more reliably and cost effectively.

The credentials that allow jobs to execute must be carefully guarded. The data moved by the automation system must also be protected. The planner needs to keep this data safe and requires a system that helps him or her secure these valuables.

The planner must also integrate into the enterprise architecture. Using systems that can communicate with the built-in interfaces provided by many enterprise applications will help capitalize on the business logic encapsulated by those systems. A single point of scheduling helps take a holistic view of the automation plans running throughout the enterprise. Deploying and maintaining those plans centrally helps control labor costs and keep plans running reliably.

A well-designed automation system will help information flow through the enterprise efficiently. It should help to minimize labor costs and keep the process reliable. A planner equipped with the correct automation system tools should be able to build a robust, effective system that makes optimal use of enterprise resources.

## 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>.