

Realtime  
publishers

# Solutions for Automating IT Job Scheduling

Greg Shields

sponsored by



Simplifying integration

Removing complexity

Shortening time to implementation



Conquer IT  
Complexity  
with Simplicity

Through  
Intelligent  
Automation

### Integrating Your Business Applications Across Distributed Computing Environments

ActiveBatch® Enterprise Job Scheduling and Workload Automation, from Advanced Systems Concepts, removes the complexity from integrating business applications, databases, and mainframes. Focused on shortening the time to implementation and improving service levels by eliminating custom scripting via its Integrated Jobs Library, ActiveBatch automates job scheduling across diverse distributed computing environments, creating a centralized view of operations at the project, organizational or enterprise level.

Learn More at [www.ActiveBatch.com](http://www.ActiveBatch.com)

---

Chapter 4: Implementing Enterprise Job Scheduling: A Requirements Checklist .....	46
Creating a Requirements Spec for Job Scheduling .....	46
Requirement #1: The Solution Shall Integrate with All Platforms and Applications that Comprise the Business System .....	48
Requirement #2: The Solution Shall Expose All Properties and Methods Used by Those Systems .....	48
Requirement #3: The Solution Shall Include Support for Script Language Independence .....	49
Requirement #4: The Solution Shall Support Queues for Job/Plan Prioritization and Performance Management .....	49
Requirement #5: The Solution Shall Support File-Based Triggers.....	50
Requirement #6: The Solution Shall Support Message-Based Triggers.....	50
Requirement #7: The Solution Shall Support Event-Based Triggers.....	51
Requirement #8: The Solution Shall Support Time-Based Triggers.....	51
Requirement #9: The Solution Shall Support Variables and Dynamic Data Across Jobs and Plans .....	51
Requirement #10: The Solution Shall Support Intra- and Inter-Workflow Communication .....	52
Requirement #11: The Solution Shall Support Job/Plan Execution via a Business Calendar.....	52
Requirement #12: The Solution Shall Support and Report on Job Object Dependencies .....	53
Requirement #13: The Solution Shall Include Client, Browser, and Mobile Device User Interfaces.....	54
Requirement #14: The Solution Shall Support Nesting, Chaining, and Load Balancing of Jobs and Plans .....	54
Requirement #15: The Solution Shall Use an Object-Oriented Management Interface	55
Requirement #16: The Solution Shall Use a Centralized Job and Script Library that Includes Commonly-Used Built-In Items .....	56
Requirement #17: The Solution Shall Consolidate Error and Output Messages from Executed Jobs .....	57

Requirement #18: The Solution Shall Support a Centralized Security Model Across Jobs and Plans ..... 58

Requirement #19: The Solution Shall Include a Centralized Change Management Model ..... 58

Requirement #20: The Solution Shall Include a Centralized Audit Database that Includes Metrics and Alerting ..... 59

So, Do You Need a Job Scheduling Solution?..... 60

## **Copyright Statement**

© 2011 Realtime Publishers. All rights reserved. This site contains materials that have been created, developed, or commissioned by, and published with the permission of, Realtime Publishers (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 its web site sponsors. In no event shall Realtime Publishers 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 book was downloaded from Realtime Nexus—The Digital Library for IT Professionals. All leading technology books from Realtime Publishers can be found at <http://nexus.realtimepublishers.com>.]

## Chapter 4: Implementing Enterprise Job Scheduling: A Requirements Checklist

---

Purchasing and implementing an IT job scheduling solution nets you only an empty palette within which you can create your own automations. Filling that palette to meet the needs of your environment is the next step.

You might remember this idea as the closing thought of the previous chapter. It highlights an important realization to keep in mind as you’re considering an IT job scheduling solution: *Once you’ve selected, purchased, downloaded, and incorporated into your infrastructure an IT job scheduling solution, what do you have? With many solutions, not much.* Once installed, some solutions expose what amounts to an empty framework inside which you’ll add your own jobs, plans, and schedules.

An IT job scheduling solution is, at the end of the day, only what you make of it. Right out of the box, a freshly installed solution won’t immediately begin automating your business systems. Creating all those “little automations” is a task that’s left up to you and your imagination.

That’s why finding the right IT job scheduling solution is so fundamentally critical to this process. The right solution will include the necessary integrations to plug into your data center infrastructure. The right solution comes equipped with a rich set of triggers that bring infinite flexibility in determining when jobs are initiated. And the right solution helps you accomplish those automations easily, carefully, and with all the necessary tools in place to orchestrate entire teams of individuals. *Integrations, triggers, and administration*—these should represent your three areas of focus in finding the solution that works for you.

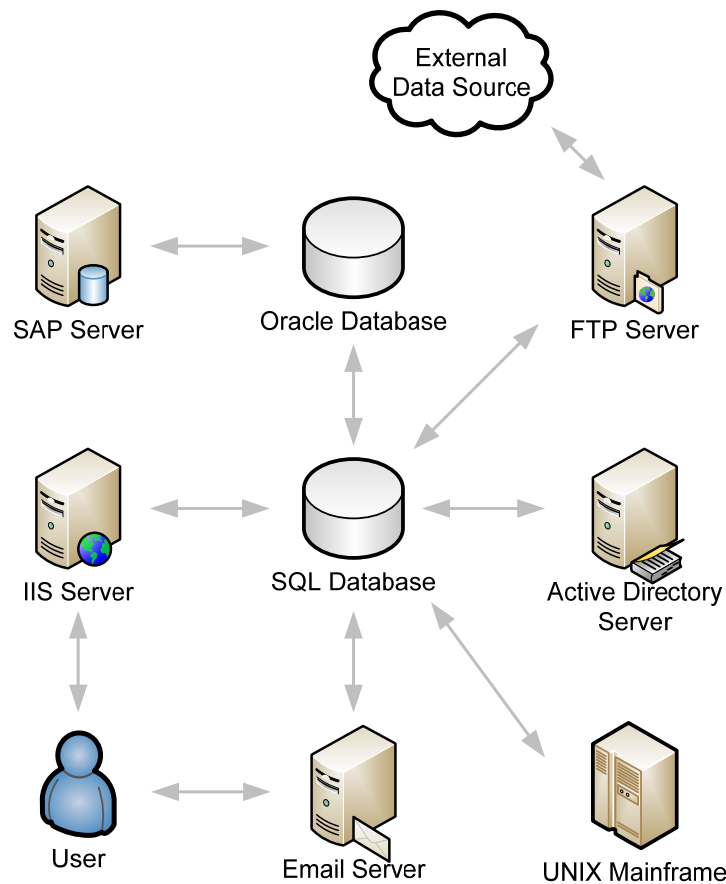
### Creating a Requirements Spec for Job Scheduling

Just three things, eh? That’s easy to say when you’re just the author of some book on solutions for automating IT job scheduling. The real world simply isn’t as cut and dried. The reality is that businesses today require justification—and often formal justification—in order to convert a tool that’s desired into a tool that actually gets purchased.

Oftentimes, IT professionals know via gut instinct that they need *something* to solve their current problem. They often even have a vague notion of what that something probably looks like. The difficult part for many is in translating their instinct into a set of requirements that lay out exactly what they need.

That's why I've dedicated this final chapter to assisting you the IT professional in creating a formal requirements specification. I'll outline a set of requirements that are remarkably similar to those used to find the solution for my project, *The Project That Would Change Everything*.

You remember that project, first introduced in Chapter 1? Its architecture is reprinted as Figure 4.1. *The Project That Would Change Everything*, as you can see, incorporated a range of technologies along with associated triggers for moving data around while processing it at the speed its business required. Finding a single-source solution to accomplish all of this wasn't an easy task. Thus, locating the solution that worked for us needed a set of formal requirements.



**Figure 4.1: The Project That Would Change Everything.**

In the next sections, I'll lay out the most important of those *in formal requirements language*. For each requirement, I'll add a bit of extra commentary to its story and, where possible, show you a mock-up of what a potential fulfilling solution might look like. You're welcome and encouraged to reuse these requirements along with their justifications in your own specification for finding the product you ultimately need.

Oh, and you're welcome. Consider these your requirements for finding an IT job scheduling solution that'll work for your needs.

### Requirement #1: The Solution Shall Integrate with All Platforms and Applications that Comprise the Business System

Not to belabor the point, but any IT job scheduling solution you select must work with every technology if it's to be useful. That means support for your databases, along with their query and management languages. It means integrating with your applications either directly or through exposed Web services. It requires direct and indirect integration with all forms of file transfer because data that's processed almost always needs to be moved somewhere else at some point. Finally, it must be able to handle data transformation, converting data between formats as it is processed or relocated.

With a sketch of the integration points that comprise your business systems, compare its list of products and technologies with those supported by the IT job scheduling systems you're considering. Those that don't support every technology should be immediately removed from your candidate list.

### Requirement #2: The Solution Shall Expose All Properties and Methods Used by Those Systems

Platforms, applications, and technologies are only the first level of integration an IT job scheduling solution requires. In addition to general support for an application, such a solution must be able to dig into that application's activities and behaviors if it is to process and move around data.

Equally as important as the support of those properties and methods is their exposure within the IT job scheduling solution. Not every business system is well documented, and not every property, method, or action has a well-known reason for being. Thus, a solution that can interrogate its integrations for what's available becomes critically important. Figure 4.2 shows how this might look for a Web Service, where a mock-up IT job scheduling solution exposes a list of potential actions and data (in this case, properties and methods) with a single click.

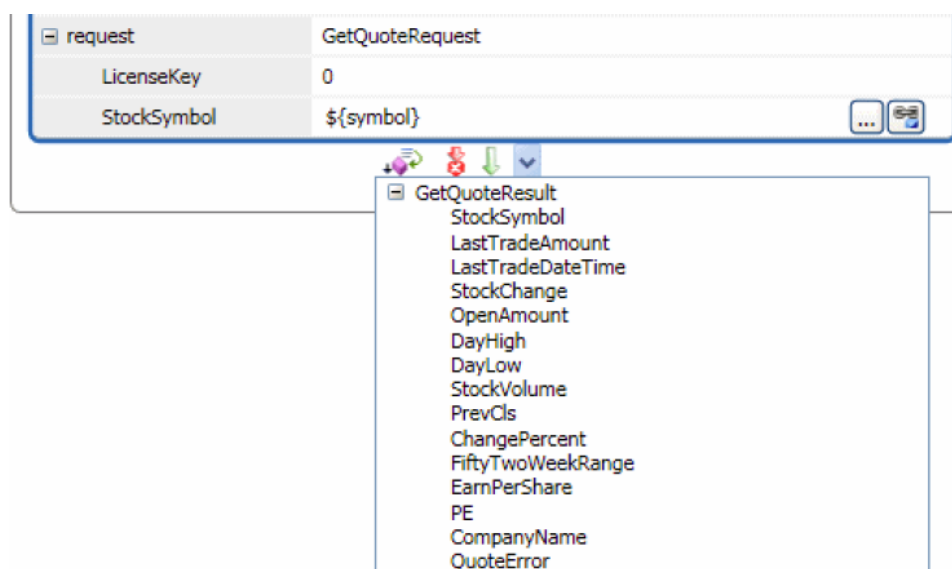


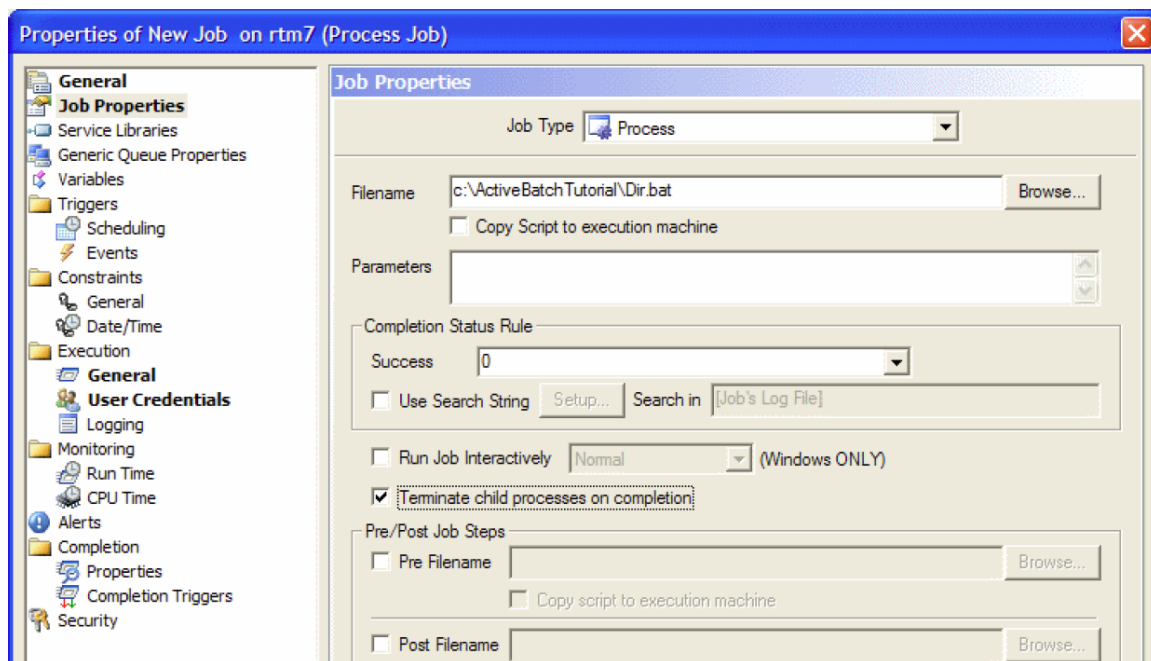
Figure 4.2: Exposing the properties and methods of a Web services object.



### Requirement #3: The Solution Shall Include Support for Script Language Independence

Orchestrating activities across platforms, applications, and technologies is only useful when the IT job scheduling solution can do so across the entire field of scripting languages. *Script language independence* refers to the requirement that any appropriate scripting language can be used within any job object and against any applicable platform, application, or technology.

Figure 4.3 shows how this might be implemented in a sample solution. Here, the job object itself does not place constraints on the type of script launched within the properties of the job. In this figure, any script can be inserted into the Job Properties location. That same script, irrespective of its language, can be further constrained via parameters, completion status, and other factors including pre- and post-execution steps. This flexibility is necessary because you'll be connecting your scheduling solution to many types of technologies, any of which may require a specific language for interaction.

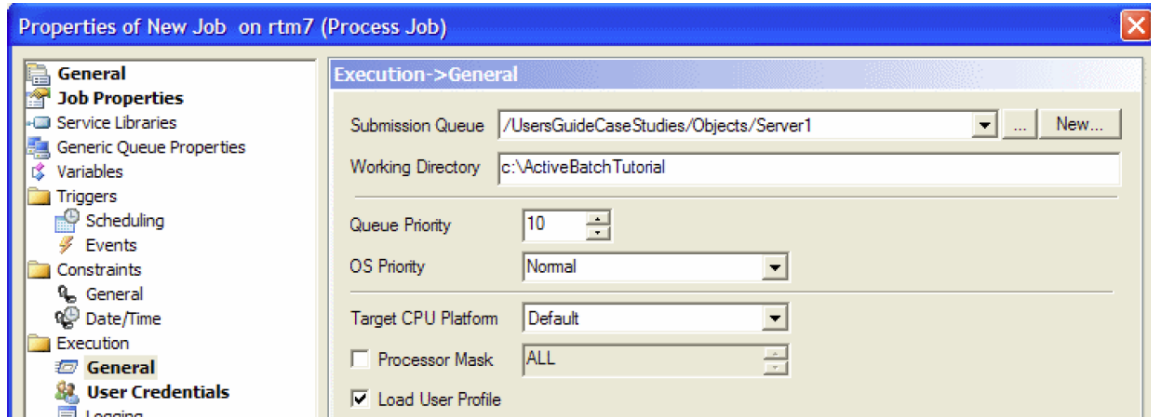


**Figure 4.3: Scripts of any language are components of each job or plan.**

### Requirement #4: The Solution Shall Support Queues for Job/Plan Prioritization and Performance Management

Queues in an IT job scheduling solution represent a mechanism to manage and prioritize job and plan activities. A fully-functioning IT job scheduling solution will leverage multiple queues of differing priorities in order to preserve performance across both the scheduling system and those it connects with. Working with a series of prioritized queues also enables a kind of failover when the resources needed by job objects are for some reason not available. In this scenario, job objects in one queue can be failed over to subsequent queues for processing. The result is a better assurance that jobs will succeed when the system experiences resource outages or other transient problems.

You can see a mock-up of how this might look in Figure 4.4. Here, a job is configured to execute within a specific submission queue. That queue is given a priority along with other parameters that control its performance. Running jobs in this manner ensures that they execute based on priorities that are driven by business rules.



**Figure 4.4: Individual jobs or plans are assigned to queues.**

#### Requirement #5: The Solution Shall Support File-Based Triggers

There's an idea in the sixth story of Chapter 2 that warrants revisiting: "Triggers are the real juice in an IT job scheduling solution. The kinds and capabilities of triggers a job scheduling solution supports makes the determination between one that's enterprise ready and one that's not much more than the Windows Task Scheduler."

It is indeed the flexibility of those triggers (along with their associated constraints) that separates the best-in-class IT job scheduling solutions from those you won't want. Requirements 5, 6, 7, and 8 all deal with the need for different types of triggers that fire based on state changes or other behaviors on target systems.

A file-based trigger initiates job execution based on the presence or characteristics of a file on a system. These triggers are particularly useful for identifying when a file is created, then firing the job's next step based on that file creation. They can do the same when files are modified, deleted, or any other action associated with that piece of data. File triggers become important for eliminating lag in distributed systems because they initiate processing steps immediately as data experiences a change.

#### Requirement #6: The Solution Shall Support Message-Based Triggers

Messaging systems such as CORBA, Java Messaging System, and Microsoft Messaging Queue among others are a low-level solution for orchestrating activities across applications and platforms. Their centralized approach to signaling across system components creates an easy framework for developers. They can be similarly easy for IT job scheduling solutions to work with.

Interrogation and integration enables message-based triggers to coordinate the activities between low-level systems and their accompanying shrink-wrapped solutions. Message-based triggers are similar to file-based triggers in that they improve job execution performance by executing actions at exactly the moment they're needed. Your chosen solution should tie into the messaging systems that are used by your business systems, enabling you to extend the reach of their signaling in and among distributed systems.

#### **Requirement #7: The Solution Shall Support Event-Based Triggers**

Like messaging systems, events are a rich source of information about on-system behaviors. With virtually every application reporting its state through OS and other onboard event systems, an IT job scheduling solution with event-based triggers gains the ability to orchestrate application events with other activities.

Most important here is the ability to customize and tailor events inside the business system. Events can be fired based on the activities within a system, so monitoring for their creation allows an IT job scheduling solution to immediately invoke resulting actions elsewhere.

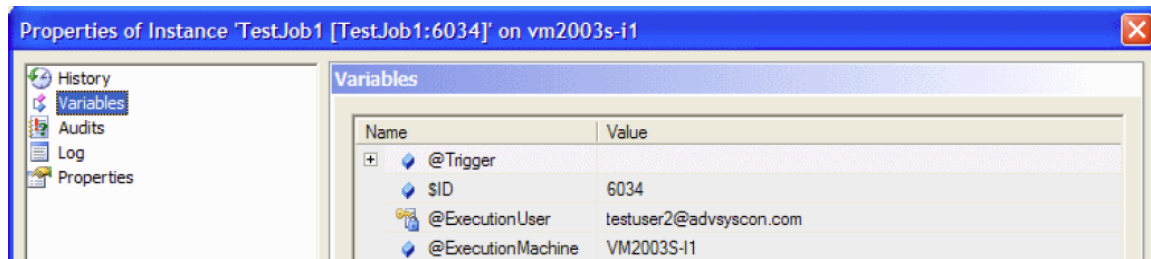
#### **Requirement #8: The Solution Shall Support Time-Based Triggers**

Although most of this book has been dedicated to highlighting why time-based triggers aren't good enough for most business systems, there still comes the time when a job must be fired based on wall clock time. Most important to recognize here is that date- and time-based scheduling can be done well (when not done well, it can be a significant limiter). An IT job scheduling solution that does not include support for multiple schedules, irregular schedules, and highly-custom schedules won't be enough for your needs, particularly in today's global workplace where jobs that span time zones may be common. Seek out those that provide high levels of customization for date- and time-based schedules.

#### **Requirement #9: The Solution Shall Support Variables and Dynamic Data Across Jobs and Plans**

Chapter 1 introduced the notion of parameterization when it comes to IT jobs and plans. This activity essentially abstracts every piece of data into variables that can be used anywhere. Variables and other types of dynamic data are critical to reusability in an IT job scheduling solution. Your chosen solution must include those that support variables both within jobs and plans as well as across them.

Oftentimes reusability of variables and other dynamic data across job objects is referred to as creating "profiles" of data. Those profiles provide an easy way to reference data no matter where it becomes needed. Figure 4.5 shows how such variables can be instantiated within a job object. There, \$ID, @ExecutionUser, and @ExecutionMachine variables have been created for later use.



**Figure 4.5: Variables are created for specific use or across all jobs and plans.**

### Requirement #10: The Solution Shall Support Intra- and Inter-Workflow Communication

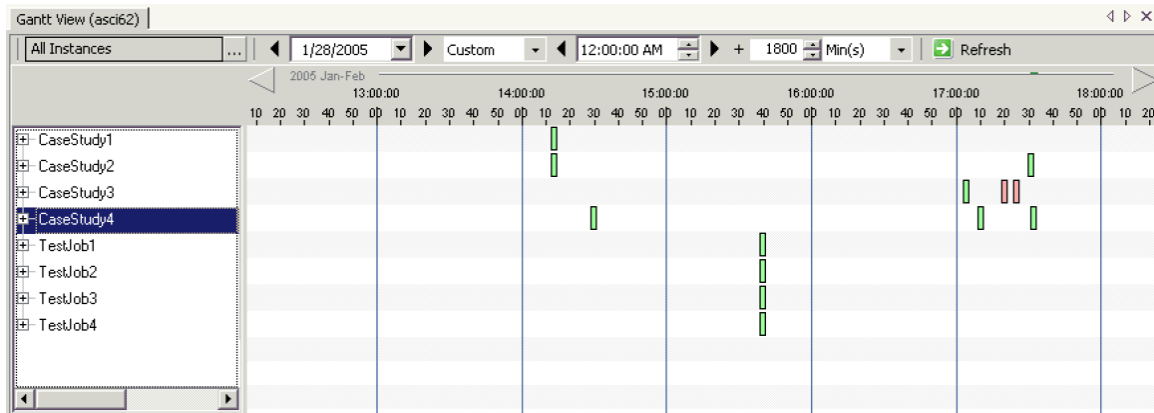
Once installed, you'll be quickly creating lots of individual jobs and plans for automating your environment. As you learned in Chapter 3, those jobs and plans are the discrete actions that ultimately connect to create a workflow. An effective IT job scheduling solution will enable the reuse of variables both within and across workflows so that very large automations can be much more easily laid into place.

You've seen in the previous chapter an example of how communication within and across workflows is useful. Its notion of exchanging data is important to simplify workflows and achieve parallelism of job processing for improved performance.

### Requirement #11: The Solution Shall Support Job/Plan Execution via a Business Calendar

You might think that a solution whose primary goal is job execution performance wouldn't need to consider the business calendar. On the contrary, it is important to recognize the impact of jobs on actual system performance. You don't want to run particularly resource-intensive jobs against production systems during periods of heavy use. Just the act of running those jobs can have a negative impact on the system as a whole.

Determining that exact "period of low use" isn't often an easy task. Business systems, particularly those that service users across multiple time zones, may experience unexpected hours of high and low utilization. The complexities of global utilization drive the need for scheduling based on a business calendar. Figure 4.6 shows one representation of how that business calendar implementation might look. Using such a calendar, the execution of entire series' of jobs can be visually identified and scheduled to prevent resource overutilization.



**Figure 4.6: Scheduling of jobs via a business calendar.**

Business calendars aren't only for resource preservation. They can also be used to schedule job activities based on when data gets created, mirroring the job execution to the business rules that drive its data. For example, if you know that a set of end-of-day data will be available at the close of each day, using a business calendar can orchestrate the collection of that data across time zones and in accordance with other business rules. By following a business calendar, it becomes possible to align the technical activities on the system with the personnel activities in the real world.

#### Requirement #12: The Solution Shall Support and Report on Job Object Dependencies

Workflows that are comprised of numerous job objects will grow unwieldy over time. This happens as individual items get ever-more interwoven throughout the greater system. Reuse of job objects at the same time creates a web of interdependencies between those very objects, which then requires management. Lacking visualization on dependencies, administrators can too easily manipulate a job object or item without realizing its downstream effects.

Figure 4.7 shows a sample report where one job object's dependencies are listed, along with their label, name, and path. This simple report is a powerful tool in a complex system where job objects find themselves reused across systems.

Associated Objects (asci62)					
ID	Object Type	Label	Name	Path	
1116	Job	CaseStudy1	CaseStudy1	/CaseStudy1	
1122	Job	JobA	JobA	/CaseStudy2/JobA	
1124	Job	JobB	JobB	/CaseStudy2/JobB	
1129	Job	JobA	JobA	/CaseStudy3/JobA	
1131	Job	JobE	JobE	/CaseStudy3/JobE	
1136	Job	JobA	JobA	/CaseStudy4/JobA	
1138	Job	JobB	JobB	/CaseStudy4/JobB	
1143	Job	BadEmailJob	BadEmailJob	/CaseStudy7/BadEmailJob	
1145	Job	FTPJob	FTPJob	/CaseStudy7/FTPJob	

**Figure 4.7: A report on the dependent objects of an object.**

### Requirement #13: The Solution Shall Include Client, Browser, and Mobile Device User Interfaces

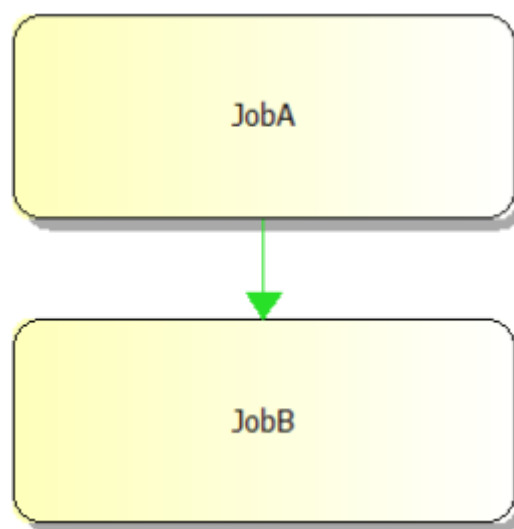
Your IT teams will find themselves growing quickly reliant on your IT job scheduling solution for more of their daily activities. That's because any action or behavior that can be characterized into a script or other job can be trivially implemented into your IT job scheduling solution.

For these and other reasons, that same solution must support multiple interfaces for management and administration. Obviously, a client-based solution will provide the richest interface for manipulating jobs and their characteristics; however, not every administrator is always in a location where that client GUI is accessible. Web browser or mobile device interfaces become valuable tools when you're in the data center and nowhere near a rich client. They become even more valuable (particularly in the case of mobile device support) when critical jobs might alert in the middle of the night. Choose a solution that includes numerous interface options, and you'll thank yourself down the line.

### Requirement #14: The Solution Shall Support Nesting, Chaining, and Load Balancing of Jobs and Plans

A large portion of Chapter 3 was dedicated to deconstructing the elements in a typical IT workflow. Each of those individual items can be encapsulated into a job or plan. Each performs some action, and interconnecting them in complex ways—such as nesting and chaining—is what makes job scheduling so extensible across the range of business services.

Figure 4.8 shows what a view might look like in an IT job scheduling solution. There JobA and JobB are graphically connected to show how results from JobA are used in the execution of JobB. Although simple in this example, the chaining of input and output represents one of the core value propositions of an IT job scheduling solution. Its orchestration of activities across all jobs and all system components is what enables this chaining to occur.



**Figure 4.8: Two jobs, the execution of which is chained together.**

The same holds true with job nesting, where the execution of one job occurs within another. Job nesting furthers the reusability of jobs by enabling an individual job to perform an action within the confines of another. Input and resulting output can be used between jobs.

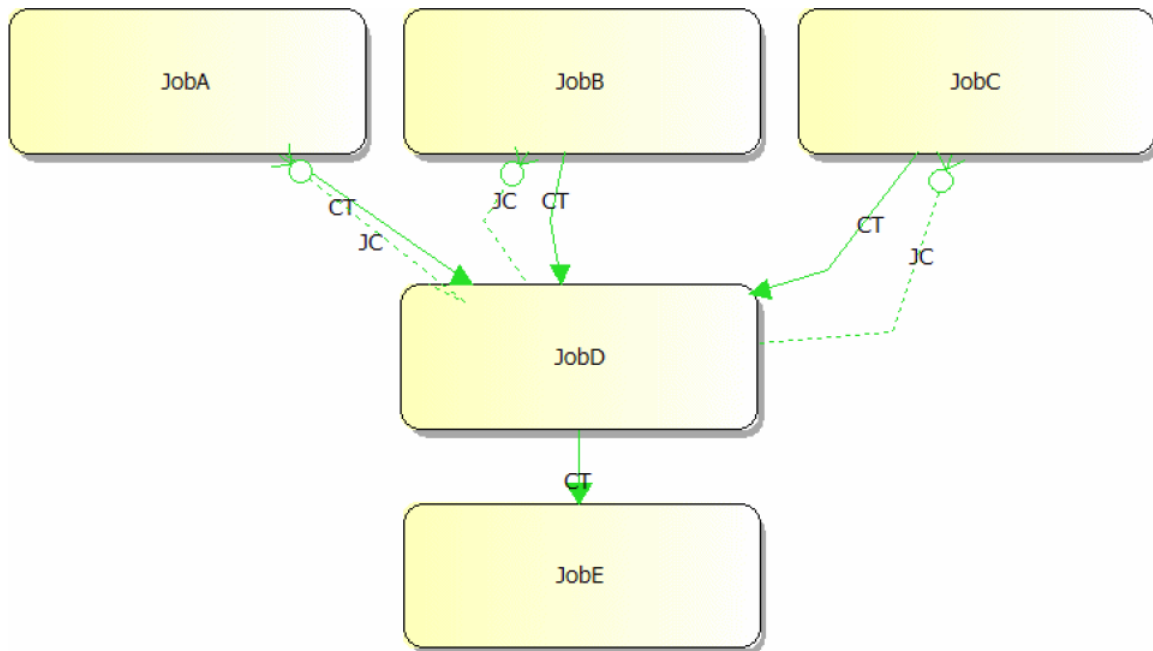
There's a third activity that's an important part of this requirement. Consolidating chaining and nesting into an overarching system highlights the power gained through job load balancing. Already discussed to some extent as a function of Requirement #11's business calendar, job load balancing enables an administrator to enact change across dozens or hundreds of system components at once. An effective solution will enable that action to occur without the fear that running a massively parallel job will impact the platform or application, or the greater system as a whole.

### **Requirement #15: The Solution Shall Use an Object-Oriented Management Interface**

One can really only get to this nirvana of complete automation if it can be properly visualized. In fact, one of the greatest limitations of many applications and platforms lies in their lack of visualization tools. You simply can't see their activities as they fire. The IT job scheduling solution you want will expose a workspace into which job objects can be laid out, interconnected, and watched as they execute.

This visual approach to job creation becomes particularly important as the scope and complexity of plans increases. As you can imagine, it's not that difficult to connect two jobs together like what you saw back in Figure 4.8. Yet the situation changes dramatically when greater numbers of tasks require orchestration, all with their own execution triggers and constraints.

An IT job scheduling solution's workspace designer functionality grows more important as complexity increases. Figure 4.9 shows what is still a relatively simple plan; this time comprised of five separate jobs. Connecting these jobs are the triggers (marked as CT) and constraints (marked as JC) that combine to determine when the next set of actions is to be executed. In this example, three jobs must coordinate their activities prior to the fourth one executing. Only after that fourth job executes can the fifth and final job complete.



**Figure 4.9: A collection of jobs in a plan with associated triggers and constraints.**

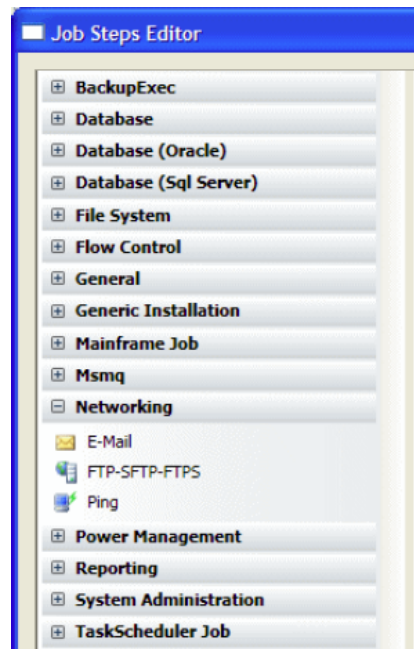
Look for an easy-to-use workspace designer tool in your chosen solution. Lacking one that presents visualizations in an easy-to-understand format makes your work more difficult and introduces the chance of failure or error in plan creation.

#### **Requirement #16: The Solution Shall Use a Centralized Job and Script Library that Includes Commonly-Used Built-In Items**

Scripts are obviously the backbone of any IT job scheduling solution, but many common actions in an IT environment are repetitive and/or easily captured into a reusable object. This chapter in fact began with the assertion that a freshly-installed IT job scheduling solution creates an empty framework that you're responsible for filling with automation. The reality is, depending on the solution chosen, that framework may be automatically populated with common actions that can be immediately usable.

As you can imagine, having a collection of jobs readily at hand can significantly speed the creation of new automations. Need to email a document? Just drag the "Email" job step into your workspace designer. Figure 4.10 shows a mock-up of how such a job steps editor might look. There, you can see a range of common activities that span the breadth of data center platforms and applications.





**Figure 4.10: An editor that includes commonly-used jobs.**

Although these job steps alone won't be specific to your needs, an effective solution will include ways of incorporating variables and other dynamic data to customize the job steps for the needs of the automation under construction. More importantly, these job steps are pre-generated and pre-tested from the vendor, which reduces the risk of scripting error and the level of effort in testing.

#### **Requirement #17: The Solution Shall Consolidate Error and Output Messages from Executed Jobs**

Easily one of the most difficult activities in creating automations is in recognizing their output, whether that be the data you're looking for or an error message. Most automations are not run in interactive mode. Instead, they're run as background processes that work with platforms and applications without exposing their activities to the logged in user. Thus, the resulting data and error messages from these scripts aren't easily captured using simple native tools.

An effective IT job scheduling solution will often execute its scripts within its own runtime environment, or within one where output and error messages can be captured. Executing scripts and other objects in this way enables the IT job scheduling solution to return this information to an administrator's console for review. Knowing output messages from executed scripts assists greatly in the generation of those scripts, easing their development process and reducing the risk of error. Look for an IT job scheduling solution that supports script execution reporting that includes output data as well as runtime error messages.

### Requirement #18: The Solution Shall Support a Centralized Security Model Across Jobs and Plans

Chapter 2's final story relayed the painful situation where a script gets misused. Script misuse, accidental use, or malicious use are all common risks in any data center environment where multiple individuals work together. That's why an effective IT job scheduling solution will include a permissions structure that can lock down jobs, plans, and even variables to specific users and/or uses.

Having a centralized security model significantly reduces the risk that a script with significant impact cannot be accidentally or maliciously run against data center equipment. It also provides a point of control for change management administrators and auditors to monitor. Data centers that operate under heavy regulation or security controls will greatly benefit from centralizing the permissions structure for script execution into a single solution.

### Requirement #19: The Solution Shall Include a Centralized Change Management Model

Security isn't the only mission-critical requirement in a solution that could potentially make massive changes across hundreds of systems at once. No less important are the needs for change control and revision history of any automations that have been introduced into the system.

You've surely experienced the situation where "something got changed." Whether that change is to a setting on a server or a line in a script, figuring out exactly what got changed—and who changed it—in this scenario is a challenging task that isn't often successful. When changes are made that inappropriately alter data, finding the exact line or character at fault adds even more difficulty.

That's why an IT job scheduling solution that you'll want to use will store revisions of scripts and other automations for review. An excellent solution will provide a mechanism for you to analyze the individual changes between revisions, as well as note which user made the change. Figure 4.11 shows an example screen where 10 revisions of a script have been logged. There, each version can be viewed to identify "what got changed."

Date/Time	Revision	Description
2/26/2009 3:16:33 PM	10	Modified by advsyscon\...
2/26/2009 3:16:24 PM	9	Modified by advsyscon\...
2/23/2009 11:02:08 AM	8	Modified by advsyscon\...
2/23/2009 11:00:51 AM	7	Modified by advsyscon\...
2/23/2009 11:00:39 AM	6	Modified by advsyscon\...
2/23/2009 11:00:30 AM	5	Modified by advsyscon\...
2/23/2009 11:00:18 AM	4	Modified by advsyscon\...
2/23/2009 11:00:18 AM	3	Modified by advsyscon\...
2/23/2009 11:00:13 AM	2	Modified by advsyscon\...
2/14/2009 12:01:04 AM	1	Created by advsyscon\...

Figure 4.11: Revision history.

## Requirement #20: The Solution Shall Include a Centralized Audit Database that Includes Metrics and Alerting

The final requirement here ties each of the last few into a centralized database for auditing, monitoring, and alerting purposes. It has been said repeatedly in this chapter that (in addition to enhancing job scheduling itself) a primary reason for implementing an IT job scheduling solution is for centralization of job execution. By default, this centralization automatically creates a single location where all actions to your business systems can be logged and monitored.

Administrator and even user alerting represent useful additions to the feature set of such a solution. Remember that any IT job scheduling solution sits in the center of your business service, orchestrating the communication and processing of data between disparate components. From this location, it is uniquely positioned to watch for and alert on behaviors in data. Those behaviors can be things of interest to administrators; or, more often, they are of interest to the users themselves. Creating alerts across all the usual alerting approaches such as email, messaging, instant messaging, and even more-modern techniques such as social media outlets provide a way to notify users when conditions of interest occur. Figure 4.12 shows a simple example of an email alert that can be initiated based on either a trigger or other preconfigured condition.

The screenshot shows a dialog box titled "Alert" with the following configuration:

- Alert Type:** Job/Plan Completed In Failure
- Delta (Mins):** 0
- Notification:** Abat.EMail
- Description:** Sends an SMTP E-Mail.
- Properties:**
  - To:** person@company.com
  - CC:**
  - BCC:**
  - Attachments:**
  - Subject:** ActiveBatch Notification from '\${@Name}'[\${@ID}]' has failed.
  - Message:** \*\*\*\*\* □ □ \*
- To:** Recipient e-mail address (i.e. person@company.com). Multiple addresses can be specified separated by a semicolon.
- Job Trigger:** <None>

Buttons: OK, Cancel

Figure 4.12: An example alert.

## So, Do You Need a Job Scheduling Solution?

So, there you have it—20 high-level requirements for quantifying the types of capabilities you need out of an IT job scheduling solution. These 20 requirements highlight the most-critical pieces that any distributed business system and its administrators will need to improve job execution performance while maintaining consistency of workflows.

And, at the same time, that's my story. In the end *The Project That Would Change Everything* was eventually implemented successfully. It took time to create the necessary automations, I'll admit. But the workflow assistance gained through the use of a centralized system ensured that all our changes were logged, monitored, and carefully categorized. In the end, given the same project and scope of work, I'd do it again in the very same way.

## Download Additional Books 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 book to be informative, we encourage you to download more of our industry-leading technology books and video guides at Realtime Nexus. Please visit <http://nexus.realtimepublishers.com>.