Realtime publishers

The Shortcut Guide™ To

Secure, Managed File Transfer

sponsored by

IPSWITCH FILE TRANSFER

Don Jones

Copyright Statement

Chapter 4: Evaluating and Selecting a Secure, Managed File Transfer Solution

In the previous chapter, I outlined some of the common business requirements surrounding managed file transfer. I explained where some of those requirements come from, and hopefully helped you figure out which requirements apply to your company and your particular situation. At this point, you should have constructed a "business requirements checklist," not unlike the one in Figure 4.1 (which is what I used to wrap up the previous chapter).

Managed File Transfer Features/Requirements List

	Importance	Supports	Notes
Encryption	5	Must comply with PCI DSS	Transferring cardholder data
Logging	5	Must comply with PCI DSS	
Hosted / SaaS	4	No staff time / resources	On-premises may be OK w/svcs
High Availability	2	Might help reduce overhead	Transfers are not time-sensitive
AD integrated	4	Single-Sign On Directive	Could also use an LDAP proxy
Java-Based Client	2	Few Mac/Unix users	Web-based client would be OK too
SFTP	4	Preferred protocols	Other similar protocols might be OK
Customization	5	Workflow is mandatory	We have minimal programming res.
Runs inside VM	4	Move to virtual datacenter	Not a concern if hosted/SaaS
Timeframe	4	Need for "Falcon" Project	Must be deployed within 2 months

Figure 4.1: Sample business requirements for managed file transfer.

The idea is that this list should contain everything that's important to your business, some notes about why they're important, and an indication of exactly *how* important they are relative to one another. I use a scale of 1 to 5, with higher numbers representing more important capabilities. This list should focus on general *business requirements,* not technical features. I didn't specify what kind of encryption is important, for example, only that encryption in general is important because my organization needs to comply with PCI DSS. I've indicated that a hosted solution is pretty important, although I'd be okay with an on-premises solution if the vendor can provide implementation services—my company just doesn't have the time to deploy a solution on our own.

Conducting Your Evaluation

These business requirements form the list that you take to solution vendors, and you get them to show you how their solution meets those needs. Obviously, every solution is going to differ somewhat in how they implement each feature that covers your business needs; part of the evaluation is to decide which implementation you like the most.

Figure 4.2 shows how I like to score solutions during an evaluation: I list all of my business needs and their importance, then rate each solution on a scale of 1 to 5, with higher numbers indicating a solution that does a better job of meeting that business need. I keep a separate list of notes with the details behind why I awarded the score I did—sometimes, as you explore different solutions, something you see in "Product X" makes you change your mind about how you viewed "Product Y," and I can go back and adjust scores as needed.

Managed File Transfer Evaluation Scorecard

	Importance	Product A Score	Product A Total	Product B Score	Product B Total	Product C Score	Product C Total	Product D Score	Product D Total	Winner
Encryption	5	5	25	3	15	4	20	5	25	A/D
Logging	5									
Hosted / SaaS	4									
High Availability	2									
AD integrated	4									
Java-Based Client	2									
SFTP and SCP	4									
Customization	5									
Runs inside VM	4									
Timeframe	4									

Figure 4.2: Product feature comparison.

Hint

You'll find a template for this kind of product comparison in drawing products such as Microsoft Office Visio and ConceptDraw PRO; you could also whip up something similar in a spreadsheet like Microsoft Office Excel, if you preferred.

Here's how I use the chart:

- Each product gets a score from 1 to 5, based on how well it supports my business need. I make pretty extensive notes that justify each score, along with explaining any subtle details I've picked up during my investigations.
- I multiply each score times the importance of that business feature, giving each product a "total" for that feature. This helps weight more important features. A product that does a great job on something that's not very important to me won't overtake a product that does a "pretty good" job on something that's mission-critical for me.
- In the last column, I indicate which product(s) "won" for that business need. The last column lets me take a quick glance and see if any one product stands out—if I see mostly "A" in the last column, then product A is probably going to go on my short list for a lab trial or pilot project.

My job in this chapter is to help you understand some of the subtle details that come into play when evaluating solutions. As I pointed out in Chapter 2, for example, all encryption is not equal; in this chapter, I'll give you some additional pointers on what that means, and what to look for when you're looking at different solutions.

Beware the Details

At a high level—just naming off features like "encryption" and "protocol support"—you'll find that most managed file transfer solutions are practically identical. Or *seem* identical. Like most technology products, these vendors know what's going to be on your feature checklist, and they aim to have everything you'll need. But each of them goes about it in a different way, and some implementations might work better for you than others.

For example, in the previous chapter I mentioned that "high availability" can be implemented in a *lot* of different ways. Some vendors might build their solution on top of Microsoft's Windows Cluster Service; others might use their own high-availability architecture. Neither one is wrong, but your company might not want to build a Windows Cluster to support a file transfer server—so that solution might not score as well for you. However, your company might have a bunch of existing Windows Clusters, and dropping a file transfer solution on to one of them might be a perfect fit—so that solution would score better with you.

It's those subtle details that make all the difference between solutions, and that's what this chapter is ultimately going to be all about. I want to emphasize that **there's never a wrong answer** in these kinds of details; there's only what's *best* for your particular situation. My goal is just to get you thinking about these details, so you can start deciding what might be best for you.

Strategic Tips

Before I start diving into feature details that you need to keep in mind, I want to cover a few broad strategic tips. As you look at different solutions, these tips are things you want to keep at the forefront of your mind at all times. In many cases, these strategic tips can change the way you view a product, help you expand your list of business requirements, and so forth.

Beauty Is Only Skin Deep

Let's face it, we all love a great-looking user interface. Most of us use Windows, a Mac, or a Unix or Linux graphical desktop, and we tend to appreciate slick-looking, graphical user interfaces (GUIs) . And there's nothing wrong with that. However, take a look at Figures 4.3 and 4.4—can you tell which of the two products is the better one?

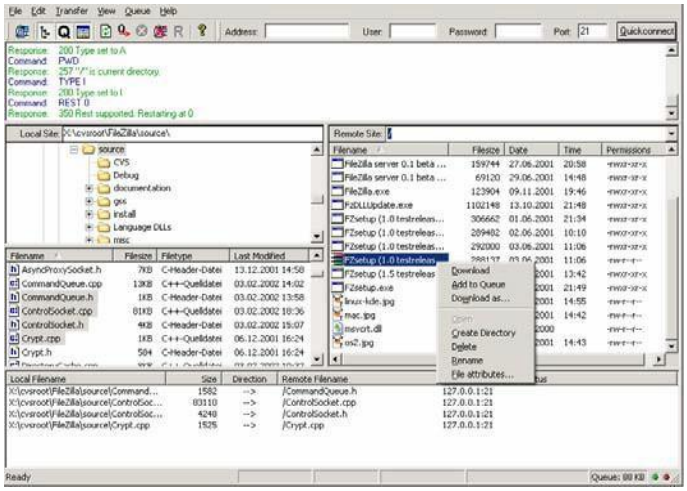

Figure 4.3: Product A GUI.

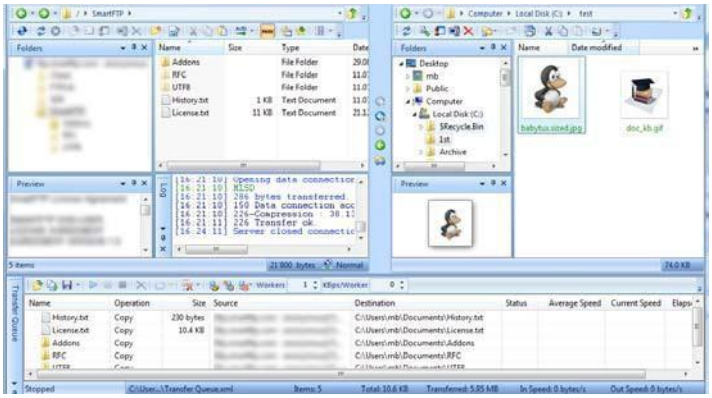

Figure 4.4: Product B GUI.

Both are graphical FTP clients, but that's not actually the point. Product B is, for many people, the more attractive of the two. It has modern-looking icons, matches the Windows XP color scheme, and so on. Product A seems like a blast form the past, with its 16-color icons and Windows 95 color scheme. The point, however, is that you *can't tell which one is better* just by looking at the user interface. What really matters is the functionality under the hood. Beauty truly is only skin deep.

This is even *truer* for a managed file transfer solution. Keep in mind that the administrative user interface in particular needs to be *functional,* not necessarily beautiful. Much of a managed file transfer solution's work is done under the hood, out of sight; *that's* the functionality you should be evaluating.

It's easy for most software developers to create a slick-looking user interface: They buy a pre-built user interface library, add it into their project, and they're done. But that tells you nothing about what's under the hood. Different vendors and different development teams often have different priorities; one team might set aside time to get an all new user interface library integrated so that their product can look just like the latest version of Microsoft Office, with whatever neat new toolbars Microsoft has cooked up for that version. Other development teams may choose to use a simpler user interface library and instead set aside more time for *server* functionality—which is the approach I tend to favor as a customer because I'm shopping mainly for function, not beauty, in a server product. Neither team is wrong, but *you* must keep in mind that you'll be living with the product's *functionality* for a long time, and you can't make any judgments about it based upon the *skin.*

Buy for the Project, Plan for the Enterprise

I have a lot of conversations with my clients about the products they're considering, including managed file transfer solutions. In almost every conversation, my client is considering a file transfer solution to solve a particular project need or to meet the needs of a particular department. Getting companywide funding for software purchases can be difficult, so they tend to respond to smaller, project-based budgets instead. That's a great tactic, but it presents a distinct problem.

As I've written before, I almost *always* see this happen: A department, division, or project identifies a need for a managed file transfer solution. In one case, it was to transfer customer orders to various vendors via secured FTP and secured HTTP. The IT department then becomes involved to help evaluate and select an appropriate solution. They properly consider every business need that the project has, and they select a solution that does an excellent job for that project. Next, another department, division, or project realizes that they, too, need managed file transfer. They come up with a list of their own business requirements, and get IT involved to find out what the company's existing solution can and cannot do.

In many cases, the new requirements don't exactly match the old ones—and in many cases, the solution the company bought for one project won't meet all the needs of another. The company then winds up buying another managed file transfer solution…and the IT department starts hating their lives because now they have to manage, support, and maintain two distinct solutions.

I always recommend to my clients that they at least *think about* other ways in which the company might use managed file transfer. You obviously can't go on an endless fact-finding mission to gather business requirements that don't yet exist, but you can use your imagination and think about other areas, and what business requirements they *might* have. As you're evaluating solutions, always ask yourself, "Does this meet the current requirements?" but also ask yourself, "Can this grow to perhaps be used as a single, companywide solution?"

By stretching your parameters just a bit, you can help avoid a situation where you wind up with multiple solutions from multiple vendors. It may be worth spending a *bit* more, for example, on a solution that offers broader protocol support and programmability, even if those things aren't strictly on your current list of requirements, simply because those things will help the solution flex to meet unforeseen business requirements in the future.

When Is Software Like a Marriage?

When it costs you money and is incredibly difficult to change your mind. Any kind of server-based, backend software solution represents a major commitment. You're going to be basing business processes on your managed file transfer solution. You're going to be teaching users how to use it. You're going to be paying money for it, and more money for ongoing maintenance. Changing your mind and switching to another solution is going to involve pain not unlike a divorce, as you migrate business processes, re-train users, spend *more* money on a new solution and on maintenance for that, and so on.

I find it's cheaper and less exhausting in the long run to err on the side of caution during your *first* managed file transfer purchase. Spend a bit more, and get a solution that *can* grow to handle needs you haven't foreseen—it'll be cheaper than redoing everything later. Spend some time researching the vendor you're thinking about purchasing from, too—because you're marrying them just as much as you're marrying their software. Ask for references from other customers. Talk to colleagues at conferences and trade shows, and see if anyone else is working with that vendor. Ask yourself some questions:

- How long have these people been in business? How stable are they? Do they have investors and money in the bank or are they teetering on the edge of solvency?
- How long have they been working with managed file transfer? Are they established veterans in the field or are they newcomers? There's nothing wrong with new players in an industry, but you need to assure yourself that they're going to stick with it.
- Do they seek out external partnerships with major vendors? Do they seek out independent testing and certification for their products? These are signs of healthy, competitive companies.

- How quickly do they respond to bugs? Do they have a history of quickly releasing "hotfixes" or do bugs always take months to fix? Talking to existing customers is a great way to get this kind of insight.
- How quickly do they respond to feature requests? Is their product development driven at least in part by what existing customers ask for or are they solely driven by what their marketing department thinks will sell new licenses? A good vendor will have a mix of customer-driven and internally-driven priorities; again, speaking with existing customers as well as with the vendor's development managers can provide insight on this.

Also try to find out a little about the history of the software you're considering. Was it developed in-house? Was it acquired from another company? If it was acquired, was the development team also brought in-house? How many people work on the software full-time? Again, there are no wrong answers here, but the answers will help you gain a feel for where the product sits in the vendor's corporate hierarchy.

Criteria for Business Requirements

Now let's start diving into some of those subtle details I've been writing about. As you review your business requirements, consider the following additional, more-specific criteria.

Security Requirements

Security is such a big driver for managed file transfer adoption, so I'm going to tackle this subject first. I have to warn you that this security stuff can get *very* detailed—that's kind of the whole point of IT security, I think—details. Yet this is also an area where some vendors occasionally "gloss over" a critical detail or two. In some cases, they do so because their product isn't *quite* on the leading edge; in other cases, it's because they genuinely misunderstand some of the fine details. Regardless, after reading the next few sections, *you'll* be able to lead the security conversation as you evaluate products.

Encryption Levels

When I first sat down to write this section, I had a whole ream of notes, covered in phrases like "256-bit symmetric keys" and "1,024-bit asymmetric keys." I even had a big argument with a colleague who asserted that 1,024-bit asymmetric keys were unbreakable, despite the fact that cryptography professor Arjen Lenstra was asked if 1,024-bit keys were dead and replied, "The answer…is an unqualified yes" (Source: http://arstechnica.com/old/content/2007/05/researchers-307- digit-key-cra ck- en dan gers-10 24-bi t-rsa.ars)."

Looking at those notes, I realized that I'm not writing a book about encryption—although I certainly could, with all this material. You probably don't need or want to become an expert on encryption any more than I want to write a whole book about it. So let's find someone more paranoid, from a security perspective, then ourselves, and see what they've done.

For me, that would be the US government. If anyone is paranoid about keeping secrets, it would have to be them. And so they had the US National Institute of Standards and Technology (NIST) whip up not only standards for encryption but also tests to determine whether a given piece of technology meets those standards. I wrote about this in Chapter 2; what NIST came up with (the most recent version, that is—paranoia is always moving forward) is a standard called FIPS 140-2. The Canadian Communications Security Establishment helps administer the testing program that goes along with FIPS, meaning you have not one but *two* governments involved—meaning the level of detail and accuracy has got to be pretty high.

For my money, I'd rather forget about figuring out encryption details and just go with whatever works for two of the world's larger governments. If the FIPS 140-2 cryptography requirements are good enough to protect "Top Secret" information, then it's good enough to protect my patient records, financial information, customer information, or whatever else. So, when it comes to "encryption" on your list of business requirements, just make sure whatever you buy is FIPS 140-2 compliance *and certified.* That "certified" bit is critical: Make the vendor prove that their product has passed the governments' Cryptographic Module Validation Program (CMVP) tests. The vendor should have a NIST certificate number. Other standards to look for include FIPS-192 for Advanced Encryption Standard (AES) algorithms and FIPS-180 for SHA-1 and HMAC-SHA-1 algorithms, some of the most important and commonly-used encryption algorithms out there today.

If a vendor can prove that they've been validated against these FIPS standards, they should get a top score in your requirements scorecard. If they can't, you're going to have to become an encryption expert to determine whether their encryption is "good enough" to meet your needs. In fact, I typically make FIPS certification a "minimum point" in my criteria, meaning I won't even talk to vendors that don't have a FIPS-certified product. Looking for FIPS certification makes my shopping easier, as the standard incorporates numerous things that I'd otherwise have to look for and evaluate on my own.

It's On You!

If encryption is something you're doing because you have an external requirement—like legislative or industry requirements—then bear something in mind: If the encryption you choose is ever compromised, the enforcing body is going to see that as *your* responsibility. "Why," they will ask, "did you not select something stronger?" However, if you've selected encryption that meets the government standard, it's hard to argue that you could have done better—and so you'll have covered your responsibilities.

Broad Security Capabilities

Review the details of your other security considerations. For example, will you need an in-solution user database, the ability to integrate with an external directory, or a solution that can do both? Most file transfer implementations involve transfers to and from external partners, so the ability to create user accounts *outside* of your internal enterprise directory can be very useful; although it's obviously convenient if your internal users can authenticate using their corporate identity, you don't necessarily want vendors showing up in your Active Directory (AD) or other directory. In some cases, companies might want the file transfer solution to have a completely independent directory and *not* integrate with other corporate directories, especially if access to the file transfer system will be limited and/or tightly-controlled.

Other security capabilities include non-repudiation. Find out exactly how each solution achieves that, be sure that you can describe to each vendor your specific reasons for needing non-repudiation, and the scenarios in which non-repudiation will apply.

Anti-Malware

It's rare, in my experience, for managed file transfer solutions to include their own anti-malware capabilities. In fact, in my opinion, it's usually *unnecessary;* few of us need *yet another* anti-malware product that needs to be continually updated and managed.

Instead, simply make sure that your existing corporate anti-malware solution will work well with a proposed managed file transfer solution. This can be accomplished through a couple of means: Sometimes, as shown in Figure 4.5, anti-malware scans occur at the corporate firewall before any data reaches the file transfer server. Firewalls may also perform scans on outgoing files, helping ensure that you're not transmitting viruses or spyware to your business partners. Bear in mind that firewall scans are usually not possible in the case of encrypted transfers simply because the firewall has no way of reading the data—that being the whole point of encryption.

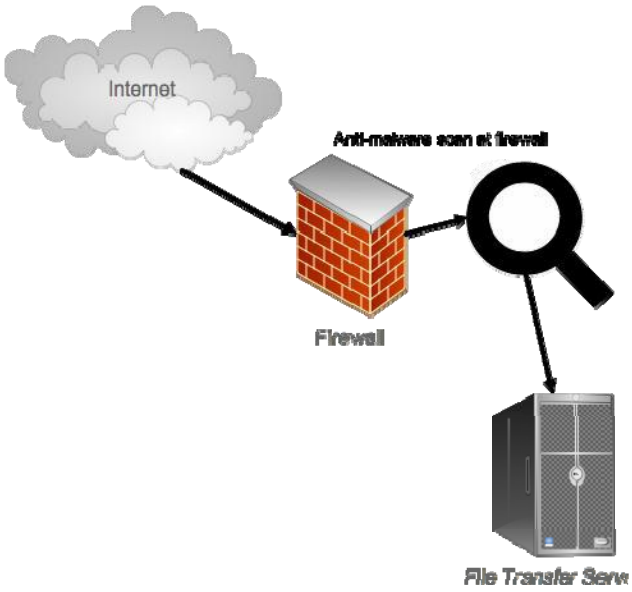

Figure 4.5: Malware scans at the firewall.

In other scenarios, you may simply install a standard anti-malware client on the file transfer server, letting that client scan files as they arrive on the server's file system—just as they would with any file server. File transfer servers typically keep files on their local file system while the transfer is in progress, and that can give an anti-malware client an opportunity to scan the file before it's transferred elsewhere. Some file transfer servers may even provide anti-malware integration points, where they explicitly request a malware scan upon completion of a transfer prior to moving the file or performing other actions. Just find out what kind of support each proposed file transfer solution offers.

High-Availability Requirements

If you've identified high availability as a requirement, do some careful research into exactly how each file transfer solution provides high availability. Again, there *are no wrong answers,* but certain techniques will be more attractive than others based on your company's experience, existing infrastructure, and so forth.

One reason that high availability can be tricky is that managed file transfer solutions need a lot of data to work; in the event of a failure, a secondary or backup server needs access to all of that data. That data includes task automation instructions, encryption certificates and keys, scripts, and statistics, and so forth. Any kind of high-availability system must involve some means for two servers to access that information. A solution built on the Windows Cluster Service, for example, accomplished this by storing information in an external drive, which is accessible to both servers. When one server stops functioning, the other can access all the needed information—but this technique has a dependency on Windows Server operating systems (OSs), compatible hardware, and the Windows Cluster Service.

Other solutions may use their own replication technologies to replicate needed information across the network to a backup server, as shown in Figure 4.6. Typically, some information is considered "private" and is not replicated—usually just basic configuration information that's handled during installation of the product. *Both* servers have the file transfer solution installed, in addition to some kind of replication service—which may be provided as an option by the file transfer solution vendor. Typically, administration is performed only against the "primary" server; that way, configuration changes can be replicated to the secondary by the replication technology.

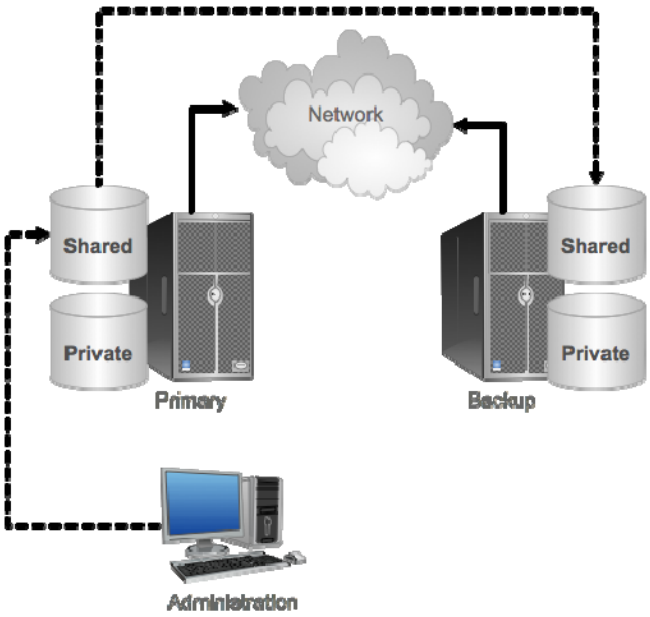

Figure 4.6: High•availability architecture.

So what happens when a failover occurs? That depends on the solution, and you should ask vendors that exact question. Here's what's ideal:

- If the primary server fails, the secondary server should pick up where it left off. In-progress transfers should be resumed, if possible—meaning part of the data replicated from primary to secondary will include the actual files being transferred.
- If the secondary server fails, the primary (assuming it's up) should continue functioning, and should queue replication updates so that when the secondary returns to service, replication can resume.

The idea is that a single server failure shouldn't impact operations for more than a few minutes, and little if any data or tasks should be lost or permanently interrupted.

Workflow Requirements

Aside from security, the need to implement workflows—review/approval cycles and other process-oriented tasks—is one of the biggest drivers behind managed file transfer adoption. If workflow is part of your business requirements, take some time to understand exactly how much work is involved in creating a customized workflow. Remember: *Technology should be driven by business,* not the other way around; *do not* accept a file transfer solution that has fixed, non-customizable process workflows. If the solution can't be made to model *your* processes, there's no reason you should change your business to model *its* processes.

Ease of Customization

How hard will it be to customize the workflow to your needs? This is the single biggest question around "workflow" as a business requirement. The following list highlights the three main techniques I've run across, in *decreasing* order of complexity:

- **Programming or scripting.** This requires the biggest investment, the most specialized skills, and the highest cost of ownership and maintenance. It can also provide the most flexibility because, generally speaking, *anything* the product can do can be reordered however you require.
- **Drag•and•drop workflow.** This is of middling difficulty and flexibility. Typically, the user interface involves something like drawing a Visio document: You drag workflow objects, like "approval" and "review" boxes, around in a flowchart-style diagram. This setup offers good flexibility but requires a bit more work. Essentially, it's "graphical programming."
- **Dialog•driven workflow.** This is typically the easiest to set up and maintain, and depending on how well it's implemented, can offer a remarkable amount of flexibility—some implementations I've seen can do anything you might imagine, in a fairly simple and intuitive interface. The usual technique has you walking through a sort of interview or "wizard," where you're asked how you want the workflow to work. Figure 4.7 shows what this might look like.

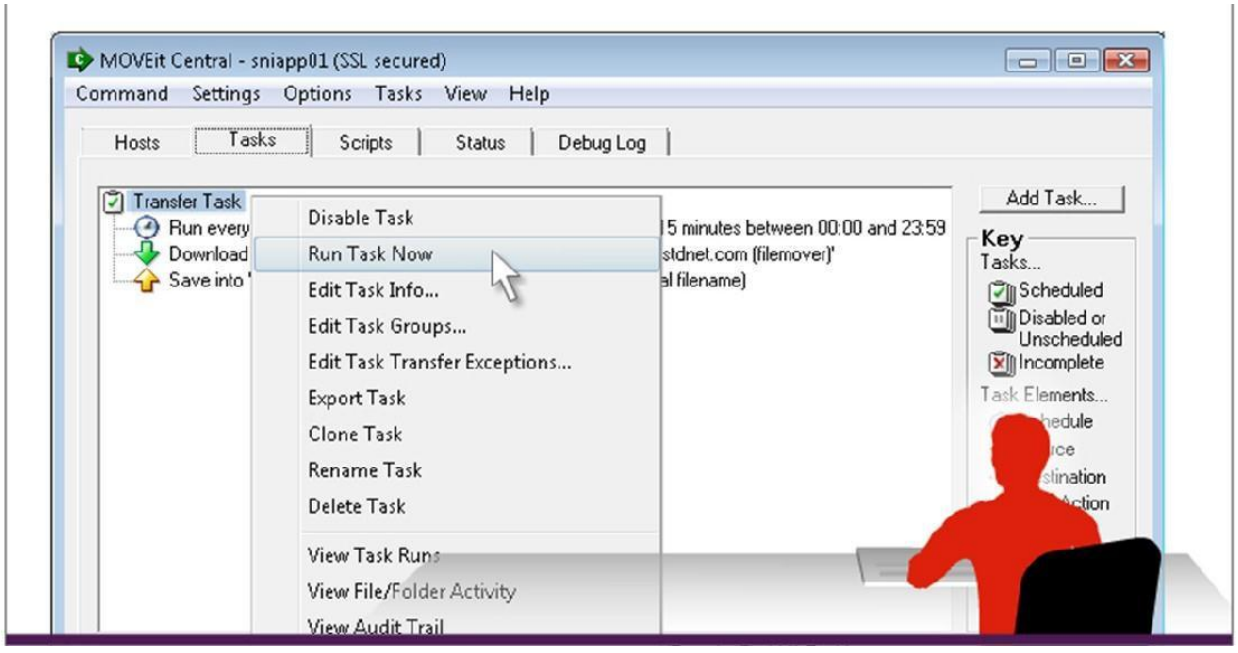

Figure 4.7: Creating a workflow using a dialog•driven interface.

The thing about "ease of customization" is this: The easier it is, the more people will be able to take on the work, and the less that will have to be dumped on already-overburdened IT staffers.

The *same* criteria apply to creating new automated file transfer tasks, such as an automated transfer of data to a business partner on a scheduled basis. This *should not* involve scripting or programming; a dialog box, or a short series of dialog boxes, should be all that's necessary to set up new tasks. Again, you don't want to *have* to rely on skilled, busy IT staffers to set up new tasks in every situation. You might *choose* to have them perform that setup for policy reasons, but the easier it is, the more easily you'll be able to find people to handle it.

Limits on Number of Tasks

Be aware that some file transfer solutions place limits on the number of automated tasks you can create. Often, limits exist in lower-priced "editions" of a product, with unlimited tasks being permitted in higher-end editions. This is neither good nor bad; it's simply something you need to pay attention to. If you need relatively few tasks, and don't have any other reason to opt for a more expensive edition, you may be able to save money by using a lower-priced edition. If you have a large number of tasks to automate, you may need to spend more. Knowing your task workload will help you and the vendor steer toward the right solution.

Always make sure that any limits on the number of tasks can be lifted, either by upgrading to a higher edition or by adding an extension or something. Ideally, you should be able to perform this "upgrade" simply by entering or installing a license key or extension; you should not have to re-deploy a whole new product. If you're looking at a solution you love, and the only way it allows you to go from a limited number of tasks to an unlimited number is to re-deploy, you might want to consider simply opting for the higher-end edition to begin with.

> **Caution**
>
> Tasks aren't the only thing that lesser-priced editions might limit, so be sure to ask about any other limitations. Remember, *limitations can save you money,* so don't opt for the high-end edition *simply* because it's unlimited. Know your needs, and buy an appropriately-sized product.

Canned Scripts and Macros

Even when a solution offers a great graphical interface for building automation and workflow, many of those interfaces produce scripts on the back-end, which is what the solution's engine uses to execute those tasks. When that's the case, having access to a library of pre-built scripts and macros can often shorten deployment and customization times. A lack of such pre-built scripts certainly isn't a deal-breaker, but it's nice to know if they're available and, if they are, what sort of capabilities they can help you achieve more quickly.

Programmability Requirements

As I described in the previous chapter, *programmability* is typically used to have *external* software run the file transfer solution. Typically, managed file transfer solutions have a number of built-in capabilities: Obviously, transferring files through various protocols is a big one, but you'll also find data-manipulation capabilities, encryption features, file compression features, and so on. The ability to access these features from within another application can be a powerful way to tightly integrate a file transfer solution into your existing business applications and processes.

Choice of API

If programmability is important to you, make sure you have—or are willing to acquire—the skills needed to work with the solution's Application Programming Interface (API). An API ties you to a specific programming language and environment; some solutions may provide multiple APIs in order to support a broader range of customers. Some examples include:

- Microsoft .NET Framework
- Sun Java
- Microsoft Windows Component Object Model (COM—accessible from C++ and older scripting languages such as VBScript, and often from within .NET Framework applications)
- Command line (usable within batch files and other system scripts)

It doesn't matter which you use—simply make sure that whatever is offered by the file transfer solution is something you're comfortable with.

Complexity of API

If a solution does include an API, how complex is it? Having your developers review the API documentation is the best way to find out. Simply make sure that your developers (or administrators because command-line APIs are often scripted by administrators) are comfortable with what they see.

If there's any doubt, speak with the vendor—particularly with a development or product manager. In many cases, vendors have built their APIs based on customer requests, and in some of those cases, vendors are willing to extend their APIs based on further requests. Helping vendors understand how and why you plan to use the API will help them understand how they might need to modify it or allow them to point out alternatives that you may have overlooked or been unaware of.

Protocol Requirements

It should probably go without saying, but let's say it anyway: Make sure the file transfer solution you select will use the protocols you need. Beyond that, however, award extra points (you're still keeping your scorecard updated, right?) for additional protocols above and beyond the ones you need because you may need additional protocols in the future. Having them built-in to begin with will save you a lot of time and effort.

Choice of Protocols

The previous chapter provided a list of basic protocols to look for:

- AS1, AS2, and AS3 secure file transfer
- Local file system and network file copy (SMB and/or NFS)
- FTP, including secure variants like FTPS and SFTP/SCP2
- HTTP, including secured HTTPS
- SMTP (email-based transfers, which I'll briefly discuss next)

Simply supporting FTP isn't enough in today's world, mainly because FTP isn't secure by default. And be *very* cautious of the "secure FTP" variants; as I described in Chapter 1, there have been *numerous* "secure FTP" attempts. FTPS (which is FTP over SSL) and SFTP (which is FTP over SSH) are currently the most popular; I've seen variations called "SecureFTP" (which was proprietary to one vendor) and I've seen other nicknames for different variations. Make sure vendors are very clear on what they support, and ask them to specify specific standard protocols, especially with the FTP variations.

"SFTP," for example, can mean:

- The SSH File Transfer Protocol, part of the Secure Shell (SSH) suite of protocols
- FTP over SSH, a normal FTP session running over a Secure Shell (SSH) connection (which provides the encryption)
- Secure File Transfer Program, a Secure Shell (SSH) File Transfer Protocol client application
- Simple File Transfer Protocol, which isn't secure at all (and which is pretty old)
- Serial File Transfer Protocol, an older protocol used over RS-232 serial interfaces

None of these are cross-compatible, of course. Of them all, FTP over SSH is common and desirable, along with FTPS, which is the normal FTP protocol running over a Secure Sockets Layer (SSL) or Transport Layer Security (TLS) connection—basically, FTPS is to FTP as HTTPS is to HTTP. I've also seen FTPS referred to as "Secure FTP." Also common and desirable is another variation of SFTP—the one that means "SSH File Transfer Protocol," which has nothing to do with "classic" FTP and is also often called "Secure FTP."

I know—it's confusing and annoying, which is why you have to spend the time to find out from vendors exactly which bits they support. Why, with all the acronyms in the IT industry, we couldn't come up with ones that involve more than the letters F, T, P, and S in different combinations, I can't tell you.

Note

There's a decent, brief discussion of the differences at http ://l00p b ack. wordp ress. com/200 9/07/ 22/ sftp -scp-ftp s-ftp -ove r-ssh-oh- the-confusion/ if you need a quick reference. There's also a quick comparison table at http://go2.wordpress.com/?id=725X1342&site=l00pback.wordpress.com&url=http%3A%2F%2Fwww.rebex.net%2Fsecure-ftp . ne t%2F which is quite accurate, and a more detailed comparison table at http://winscp.net/eng/docs/protocols#protocol_comparison.

Email as a Transport Mechanism

I like managed file transfer systems that include support for email (SMTP and POP3, usually) as a transport mechanism. Now, email isn't the most secure thing in the world whether it's encrypted or not, but *everyone* in the business world has an email address, so sometimes the broad availability and accessibility can override the less-than-secure nature of email (which I discussed at length in Chapter 2).

I commonly see email used in person-to-person transfers, where one user will use their managed file transfer system to create a secure email to an external user. Sometimes, some business partners may *only* be able to deliver data via email, so your file transfer solution has to actually log into a mailbox periodically, check for new files, then process them—that's where POP3 support comes in. Again, these aren't the optimal ways to move files from place to place, but sometimes you just *have* to, so I won't usually recommend a file transfer solution that doesn't support them.

Operational Requirements

Finally, your last set of requirements should focus on your own internal operational requirements. How well can a proposed solution be managed over the long term? Will it support other process and business requirements you may be subject to?

Audit Logging and Reporting

Call it "auditing" or "logging," too, if you prefer, but most companies—typically as part of a security effort—will want a managed file transfer solution that provides detailed logging. However, you need to be precise about what you want captured. Explain your security requirements—even if that explanation is simply, "we have to be Sarbanes-Oxley compliant," and let vendors help you understand how they achieve that.

Some specifics to look for:

- *Every* action in the file transfer solution should be logged or have the ability to be logged if logging is enabled. That includes configuration changes, task changes, file transfers of any kind, receipt of files, and so on.
- You might want troubleshooting-level logging, which means logging connection attempts and other low-level functional details.
- Log files—at least the ones related to who did what and when—need to go in some kind of secured, tamperproof or tamper-evident database. Nobody should be able to clear or alter that log without having to jump through numerous hoops; you don't want someone covering their tracks by dumping the log.
- Reporting can be incredibly useful and can help make a giant pile of log entries into something more useful. Reports may even be available for specific compliance efforts or for specific security scenarios. If reporting is a feature you need, also consider whether you need the ability to create custom reports.

Note

File transfer solutions that use an external database—such as Microsoft SQL Server or Oracle—offer an advantage in that the database can be accessed by any kind of reporting tool you might have. You'll obviously have to do more work to create custom reports in this fashion, but if you have extremely detailed reporting requirements, it's good to have that flexibility.

Monitoring

How will you monitor and maintain the file transfer solution? Some solutions may offer their own management console that includes monitoring capabilities, and in some environments, that may be all you need. Other environments may prefer to integrate the file transfer solution with an existing enterprise monitoring framework, such as IBM Tivoli, HP OpenView, Microsoft System Center Operations Manager, and so on. File transfer solutions may integrate directly with some frameworks or may integrate more generically through a common protocol like the Simple Network Management Protocol (SNMP).

Don't think that there's no need to monitor a file transfer solution; there is. You need to be alerted to problems, proactively alerted to pending problems (like low disk space), and so on.

Other Considerations

If you haven't already done so, start creating a list of usage scenarios for a managed file transfer solution. Describe, in narrative form if possible, some of the business situations where you would see file transfer coming into use. You might find that these situations—when you really start to think about the details—will help crystallize additional requirements, and you can add them to your list as you evaluate products. For example:

> We have hundreds of subcontractors who need to transfer files to us. These files consist of in-progress and completed work, and need to be transferred securely. We need to ensure that the transfer is encrypted, and that only authorized subcontractors can log in. Our subcontractors do not (and cannot) have Active Directory accounts in our domain.

This scenario reveals business requirements that you've no doubt already thought of: encryption, the ability to maintain an independent user database, and so on. But *really* think about how this might work in the real world. Will subcontractors have to download a special piece of client software in order to transfer files to you? Or will they use a Web interface? If they're using client software, what OSs do you need to provide a client for? As you conduct your evaluation, you don't need to have answers to these questions, but you *should* be asking these questions of the vendors you work with. How would their product support this scenario? In the end, you may be happy with several widely divergent answers from different vendors, which is fine—but the important thing is that you'll know *how* each of them supports this important business scenario.

> **Note**
>
> This example really stresses the need to *talk* to solution vendors. A lot of IT people would prefer to just find the information they need in a search engine or on a vendor's Web site—and you can certainly get a *lot* of information that way. But when you start diving into subtle, situation-specific details, there's nothing like a phone call with someone who knows their product. I once had a scenario similar to the previous example and couldn't find *anything* about it on the vendor's Web site. Turns out they had an entirely separate product that supported this exact scenario and I just hadn't realized it. A day or so of investigation could have been replaced by a 10-minute phone call.

Thanks for Reading

Well, that's it. Hopefully, I've helped you understand why secure, managed file transfer is such a great thing to have in your environment—how it can help offer you the security, the automation, and even the cost savings that you need. We've busted some myths about file transfer, and I've helped you figure out exactly which file transfer features are most important to you. In this chapter, I walked you through some of the things you'll need to consider when evaluating solutions, and I hope that you'll be able to really focus on some of the fine details and subtle differences when you get into your evaluation.

Corporate file transfer has come a long way from your basic FTP server; today's managed file transfer solutions offer better manageability, true workflow, powerful automation, and more. They offer a better cost of ownership than homegrown solutions, while bringing a set of capabilities that most companies really need today. Good luck with your managed file transfer projects.