Controlling Costs, Improving Compliance, and Streamlining Integration with Managed File Transfer

The Essentials Series

Dan Sullivan

Realtime
publishers

Copyright Statement

Realtime
publishers

Improving Application Integration with Managed File Transfer

There are many types of processes in a typical IT operation. Data is collected, analyzed, and reported. Transactions are recorded. Workflows move data through well-defined steps of business processes. One common aspect of all these operations is the need for data exchange to implement application integration. In many cases, this exchange will depend on file transfers. How you design and implement these file transfers can significantly impact the cost and effectiveness of business operations. It can also affect the types of applications and workflows you can implement.

Managed file transfer is a set of practices and applications that enable a reliable, consistent, and secure exchange of files in ways that meet the needs of a wide variety of business requirements. To understand the impact of managed file transfer on application integration, it helps to understand:

- How application integration enables business innovation
- Challenges to application integration
- Methods for streamlining integration with managed file transfer

There are many business drivers to adopting managed file transfer solutions, such as cost control and compliance. However, you should not underestimate the value of managed file transfer to enabling business innovation.

Application Integration Enables Business Innovation

Consider a relatively simple database that stores data about customer transactions. Typical transactions include sales orders, purchase orders, line item details about orders, and payment history. The database supports billing, order tracking, and management reporting. This type of back-office application is essential for a business to operate. It is also an untapped resource of business value.

Making Better Use of Data

There is nothing wrong with the back-office application just described. It does what it is designed to do: track customer orders. The data managed by this application has other uses as well. For example, customer transaction data can elucidate differences in customer buying patterns. For example:

- Some customers might tend to make substantial purchases early in a quarter with occasional small orders late in the quarter
- Customers might often purchase the same group of two or three products together
- Customers in different regions of the country tend to have different seasonal buying patterns

These kinds of insights are not necessarily obvious in reports listing sales by customer or time to fulfillment. Most transaction-oriented systems are not designed to analyze large sets of data looking for patterns across subsets of customers or products.

Enabling a Comprehensive View of Business Data

A different type of application is needed for a more comprehensive view of customers, markets, production, and other business operations. Data warehouses, data marts, and other business intelligence platforms are better equipped to enable in-depth analysis. They are designed to store and retrieve large volumes of data integrated from multiple applications.
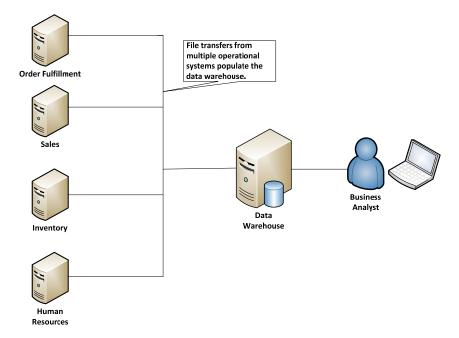

Figure 1: Managed file transfers enable organizations to build analytic applications for better insight into customers, operations, and markets.

A data warehouse is just one example of the type of system that can be created when data can be reliably exchanged between applications. There are, however, substantial challenges to application integration.

Realtime
publishers

Challenges to Application Integration

There are many ways to organize, integrate, and analyze business data, but businesses might not be taking advantage of these opportunities. The reason is at least partially due to the difficulties encountered trying to reliably transfer data between applications:

- Frequently changing requirements

- Well-known disadvantages to custom scripts

- Limited developer resources

Frequently Changing Requirements

It sometimes might seem that no sooner is an application developed and deployed than users start requesting changes. This reality is understandable. Once we work with a system, we see the types of information it provides, we understand how to work with the application, and we envision additional functions we could accomplish with it. Good analysts can look at a single set of data from many different vantage points. When they do, the can imagine what they could do if only they could add more data. More data means more file transfers and changes to the data collection process.

If file transfers are implemented using custom ad hoc scripts, frequent changes to requirements can become a significant problem for developers. This is not to say developers do a poor job of writing file transfer scripts. They do, however, tend to write as much as needed to get the job done in the amount of time they have. Good developers can see the way they would design and write their ideal file transfer program. They can also estimate how long it would take to write the ideal script. In the end, good developers have to balance the number of features included in their programs with the time they have to develop and test them.

Well-Known Disadvantage of Custom File Transfer Scripts

When developers are constrained by time and other resources, the rational approach toward development is to meet the most important and immediate needs. If the top business requirement is copying a set of files from source databases to a data warehouse staging area, that core functionality is the first thing developers would develop. It also means that other features will be added only if there is time. As a result, features that might be important at a later point are not addressed. The impact of this approach results in well-known limitations of custom file transfer scripts:

- Poor scalability

- Lack of reusability

- Code that is difficult to adapt to new requirements

- Potentially poor security features

- Limited logging and reporting features

Realtime
publishers

These limitations can be addressed when they become problems, but that can be disruptive to normal business operations. For example, if a file transfer script is not designed for scalability, as the volume of data grows, the file transfer process will take longer to complete. At some point, the script might not finish fast enough to complete data loads in the allotted time. Now scalability is a top priority and developers are challenged to go back into their code to try to find ways to optimize it. This kind of scenario leaves developers with limited time to address this new problem while addressing demands for other changes in scope or new applications.

We need a better option.

Streamlining Application Integration with Managed File Transfer

To improve application integration, we need a platform for file transfer that is readily configured to meet new requirements and provides commonly needed features, such as error handling, security, and logging. The following sections highlight features to consider when evaluating a managed file transfer solution.

Configuration-Driven Specification of File Transfer Process

One of the reasons custom file transfer scripts are problematic is that they require skilled developers to implement commonly needed functions. A better approach is to have a file transfer framework that can be configured by choosing parameters from a set of options. For example, rather than having to code a new logging routine for each new file transfer script, a configurable platform could provide three different levels of logging. The user selects the appropriate level and specifies a directory to write the log instead of writing Perl or Visual Basic code to write data to a file.

Reliable and Scalable File Transfer

When transferring large files, you run the risk of insufficient disk space, corrupt data blocks, and other problems that can impair the file transfer process. One of the ways to deal with these risks is to use intermediate checkpoints such as staging areas. You write data to disk and perform a set of data quality control checks and if the data is correct, you copy the data again to the target destination. You perform these extra steps to reduce the risk of transferring corrupt or partial data into another application.

A better approach is to incorporate checks and controls in the file transfer process to avoid unnecessary and time-consuming extra operations. For example, rather than write a file to disk, you can use file and data block integrity checks prior to writing data to the target to verify that no data is lost or changed in transit. If there is a problem and data integrity cannot be verified, the block or file can be retransmitted. Managed file transfer solutions include checkpoint/restart capabilities, eliminating the need for you to implement a potentially complex set of code.

Realtime
publishers

Another potential problem is that a file transfer will fail due to an unforeseen problem, such as a network timeout. A managed file transfer solution should be configured to monitor for changes in the conditions that caused the failure and restart the file transfer when the problem is resolved.

Management reports and alerts are essential features of a managed file transfer solution. Monitoring requirements will vary, so the solution should allow for different levels of reporting and alert generation.

Users should be confident that their file transfer jobs will not be altered. Be sure to look for access controls to limit the ability of others to change existing jobs or create new file transfer jobs.

Summary

Application integration and data exchange can enable innovative ways of working with information. Applications are often designed to perform a limited set of operations, but the data these systems collect might be used in a wide variety of ways. Managed file transfer solutions can enable reliable and scalable data exchange that is essential for leveraging the maximum value from your data resources.

Realtime
publishers