Recommended Practices

When it comes to running an FTP site the best method that we've have found is for your users to connect to your server and pick up or drop off files (take a look at our instant SFTP site). When you try to push or pull files you to have to account for the maintenance schedules and unplanned outages of your Trading Partners servers.

With hundreds or maybe thousands of Trading Partners the chaos that can ensue after an outage, planned or unplanned, becomes a full time occupation. Now we're not saying that you should never push or pull files but you should do your best to put the responsibility for file exchange on the Trading Partner. After all you are handling the file, routing it to the destination in the format desired and archiving it for posterity. To do that for all of your Trading Partners requires consistency on the exchange part of the process such that it doesn't become a process bottleneck.

Don't make yourself responsible for figuring out if the other side is up or not and don't run your business on scheduled tasks expecting the other side to have their files in place on time, they won't. Instead make your tasks respond to trigger events such as file arrival rather than expecting the file to always be there at a certain time, it won't always be there on time. Configure your Events so that IF a file or files haven't arrived by some specific time or Event occurring notify someone.

In order to download and use any of our workflow processes you must first register as a member by finding and clicking the Join Us link and you will be able to access the Downloads section of the Vortal.

To get started begin at the beginning…

Review and download the TPR package on the Downloads page link named Event Driven XML Guided Workflow. This is the base line TPR required to make use of any SkunkWorkZ developed processes. It makes use of an XML file that controls and directs the entire transfer process from file arrival to completion. Only an XML file needs to be configured to perform such task as:

1. Collect some number of files then ZIP and PGP encrypt them and send securely to a Trading Partner with email notifications
2. Poll a Trading Partner FTP server for files, pull them, decrypt and unzip then make available to the consuming application
3. Pull apart email messages sent to a monitored mailbox and make the parts of the message available to running TPRs
4. Route X12 EDI transactions to their respective folders for handling based on ISA, GS and ST header contents
5. Create an instant secure FTP site

Are my files ready yet?

A question every FTP site asks of their Trading Partners FTP site many times per day. Here we present a technique for signaling between Trading Partners that files are ready to be picked up or dropped off that comes in handy for large files.

First of all ZIP the file. This serves two purposes:

1. Compress the file so it takes less time to send and less space to archive
2. When decompressed on the other side if it is not well-formed it will fail giving the receiver a very definite error that the file is incomplete.

Second of all PGP encrypt the file

1. PGP encrypt the file so it is protected in transit and at rest on your FTP server and the Destination FTP server.
2. Here at The SkunkWorkZ Online we store our Inbound and Outbound files PGP encrypted in a SQL Server 2014 FILETABLE so they're available thru SQL statements but more importantly as separate files that can be accessed as part of the file system.

Easiest to implement is this 'semaphore' technique whereby the 'sender' deposits the file(s) in the Source folder and also drops a 'flag' or 'semaphore' in the folder of a specific name pattern that the Trading Partner firsts checks for the presence of and IF the 'semaphore' is present exchange the files.

When done delete the 'semaphore' so the other side knows they can continue dropping files in the Source folder. This is mostly effective when exchanging gigantic files that require hours to transmit. We know of sites that routinely exchange 35 GB to 75 GB files that take a long time to be posted to the Source folder.

We've also encountered problems with large mainframe FTP processes in that some instances will write a large portion of the file to the Source folder then close the file, reopen it and append the rest. Well the act of closing the file in a lot of cases indicates to the receiving process that the file is ready to be transferred so you end up with only a piece of the file. Yet when you look at the file that is completed on the mainframe process it is intact. A 'semaphore' process between the mainframe and the FTP server can rectify that problem.

In the XFT platform we've already addressed this issue and provide as part of the package the solution in source code format.

File Naming Conventions

At The SkunkWorkZ Online we require our files to follow a pattern whereby the last two nodes of the filename prior to the suffix are of the format _ccyymmdd_hhmmddss and we prefer that underscores be used as the node separator for filenames. We also expect only one period separating the last node of the filename from the suffix.

In the XFT platform we expect this request to be honored and that exceptions be handled individually and we'll provide as part of the package the solution in source code format. Not that any alerts will be generated or any functionality suppressed but you run the risk of 'stepping' on files if they aren't named uniquely.

How do you setup SSL credentials for EFT transfers over the web?

If you plan to use the Web Based or the Plain Text Client in GSB you'll want to obtain a valid CA issued SSL certificate from a large well insured Certificate Authority and get the one that provides the most insurance coverage as possible.

First go into GSB and generate the certificate request.

Second submit the certificate request to the Certificate Authority of your choice.

Thirdly install the certificate from the CA in EFT.

Details to follow…

Event naming pattern in GSB

In order to better manage turning on and off Event Rules we choose to use the COM API provided with the GSB product. The COM API allows an external program written in VB Script for instance to inquire of and make changes to an executing instance of GSB. Rather than having to logon to the GSB administration module and selectively turn on or off Event Rules the use of the COM API allows us to quickly turn on or off said rules.

Event Rules should be named such that the function is defined or at least hinted at. Try

ET_Appl_SubAppl_S where:

ET = File Monitor or SJ = Scheduled Job (Event Type)
Appl=Application moniker
SubAppl=Sub Application moniker
S= P for production T for Test

By naming the Event Rules with consistent values you can turn on and off the Event Rules from a VB script using the EFT COM API. Details and example in .ZIP format

Duplicate file prevention

Not a problem for EDI users due to the internal enveloping of EDI files (self-routing documents) but for non-EDI or Flat File handling it can be a problem. We use a simple technique here at The SkunkWorkZ Online in that we have a folder called HISTORY. In this folder is a file named the same as every file we've ever successfully pulled and processed on our side of the transaction. Inside that file you can keep other information about the transfer such as date/time started/ended, file size etc.

To determine what files to download from a Trading Partners FTP server we first obtain a list of all files in their Source folder. Then we check the list filename by filename against the names in our HISTORY folder and if a file of the same name is found we skip downloading it. Otherwise we download it, pass it along to the application that will consume it and create a file in the HISTORY folder of the same name. This prevents it from being downloaded again.

A side advantage of this technique is that to download a file again just delete it from the HISTORY folder and the next time the process runs it will come down.

In the XFT platform we've already addressed this issue and provide as part of the package the solution in source code format. In the XML for the transaction you will see two lines of XML data related to implementing this functionality:

<TP_HistoryFolder>C:\EFTRoot\Test\BM\History\</TP_HistoryFolder>
<TP_DuplicateChecking>YES</TP_DuplicateChecking>

When a file arrives for processing the process looks in the folder identified as **<TP_HistoryFolder>** for a filename that matches the filename that just arrived. IF one is found it means that file has already been processed, skip it.

ZIP'ping and UnZip'ping

We highly recommend that you ZIP your files prior to sending them and that you UnZIP your files in order to consume them. FTP is a delivery service and with the GSB implementation you receive a significant amount of Workflow capability, make use of it. Straight out of the gate you should make this a policy. You could also add a PASSWORD for added security.

In the XFT platform we've already addressed this issue and provide as part of the package the solution in source code format. In the XML for the transaction you will see three lines of XML data related to implementing this functionality:

<TP_Zipped>YES</TP_Zipped>
<TP_ZIPFolder>ZIP\</TP_ZIPFolder>
<TP_ZIPpassword>TheWord#1</TP_ZIPpassword>

When a file arrives for processing the process looks in the folder identified as **< TP_Zipped >** for a value of YES and IF one is found it means that file needs to be ZIP processed.

PGP Encryption and Decryption

We highly recommend that you PGP encrypt your files for transmission and that they sit in your file system encrypted in PGP format. Protection against snooping on the COMM line is pretty well protected against these days but most 'hacks' today occur because the hacker got by the frontend software to gain access to the data files. If your files are encrypted and a hack does occur odds are in your favor they'll be unable to use the data.

In the XFT platform we've already addressed this issue and provide as part of the package the solution in source code format. In the XML for the transaction you will see five lines of XML data related to implementing this functionality:

<TP_PGPencryption>NO</TP_PGPencryption>
<TP_PGPDecryption>NO</TP_PGPDecryption>
<TP_PGPprivateKey/>
<TP_PGPpublicKey/>
<TP_PGPpassPhrase/>

You still need to use the EFT Admin tool to inform EFT about the Keys.

How do you handle constant values?

We have a TPR that runs whenever an FTP site is started that sets values from an XML file as system level Environment Variables that are accessible to running TPRs. Each server node needs to implement this process if used.

In the XFT platform we've already addressed this issue and provide as part of the package the solution in source code format

Go to the Downloads section of the Vortal and look for System Level Constants and Properties to see HOW we set Environment Variables to use as Constants.

Can you have multiple TPR libraries?

Yes, the TPR library paths are maintained in the Environment Variables (see above) and are accessible to running TPRs and can be used in resolving substitutable arguments.

In the XFT platform we've already addressed this issue and provide as part of the package the solution in source code format

Yes you can. Go to Go to the Downloads section of the Vortal and look for System Level Constants and Properties to see HOW we set Environment Variables to use as Constants. We set a Constant value as the TPR library path and use Substitutable arguments to substitute the variable value for XML entries at runtime.

What are substitutable arguments?

The XFT platform is driven using XML files (hence the **X** in **X**FT) and rather than having to hard code literals and paths into the XML files you can use substitutable arguments to cause argument strings to be replaced by values (such as TPR library) at runtime.

In the XFT platform we've already addressed this issue and provide as part of the package the solution in source code format

```
<?xml version="1.0" encoding="utf-8"?>
<TP>
<TP_Debug>OFF</TP_Debug>
<TP_Debug_Email>TomF@SkunkWorkZ.DE</TP_Debug_Email>
<TP_Source_Mailbox>FTP</TP_Source_Mailbox>
<TP_Moniker>%TP_Source_Mailbox%</TP_Moniker>
<TP_Direction>Inbound</TP_Direction>
<TP_Source_Folder>C:\EFTRoot\SFTP\Prod\%TP_Source_Mailbox%\In\</TP_Source_Folder>
<TP_Source_Node>8</TP_Source_Node>
<TP_Source_Folder_Node>8</TP_Source_Folder_Node>
<!-- Standard E-Mail Communication Artifacts -->
<TP_Destination_Folder>C:\EFTRoot\EDI\Test\TheBucket\</TP_Destination_Folder>
<TP_Destination_File>*.*</TP_Destination_File>
<TP_EmailFrom>%_XFT_From%</TP_EmailFrom>
<TP_EmailOnError>TomF@SkunkWorkZ.DE</TP_EmailOnError>
<TP_EmailOnSuccess>%_XFT_To%</TP_EmailOnSuccess>
<!-- Folders for archiving & tracking duplicate files -->
<TP_ArchiveFolder>%_XFT_%TP%_ARCHIVE_Path%%TP_Source_Mailbox%\In\</TP_ArchiveFolder>
<TP_HistoryFolder>%_XFT_%TP%_HISTORY_Path%%TP_Source_Mailbox%\History\In\</TP_HistoryFolder>
<TP_BackupFolder>%_XFT_%TP%_BACKUP_Path%%TP_Source_Mailbox%\BackUp\In\</TP_BackupFolder>
<TP_DebugFolder>%_XFT_%TP%_DEBUG_Path%%TP_Source_Mailbox%\BackUp\In\</TP_DebugFolder>
<TP_Manifest>Manifest.log</TP_Manifest>
<!-- Transaction Processing Routines -->
<TP_Email_TPR>%_XFT_%TP%_SCRIPT_Path%SFTP_FTP_WF_Send_Email.aml</TP_Email_TPR>
<TP_Current_TPR/>
<TP_Prior_TPR/>
<TP_Next_TPR>%_XFT_%TP%_SCRIPT_Path%SFTP_FTP_FTP_TP_In_Out_MOVE.aml</TP_Next_TPR>
</TP>
```

Refer to How do you handle constant values? for detailed information on just WHERE all those values surrounded by % come from.

Monitoring SQL Server FILETABLE for Events

SQL Server has this powerful yet underused ability to store data in the database but also share it through the Windows file system. If you drop a file in a folder known to SQL Server as a FILETABLE folder it is cataloged and made available through SQL statements as well as programs that consume Windows files.

Unfortunately GSB does not see File Create, Delete and Rename events in a folder created as a SQL Server FILETABLE. It's part of SQL so it shouldn't be expected to be covered by all file system rules so we must compensate, in order to 'monitor' new files arriving in a FILETABLE we set up a Folder Monitor Event Rule that 'listens' for files arriving in some NTFS folder that handles the file processing. We then set up a Scheduled Event Rule that runs at some set interval and MOVES any files from the FILETABLE folder to the one being monitored thus 'fooling' the system into handling the file.

Combining FILETABLE storage with Zipping and PGP'ing files provides a neat way to manage backup of data files since it's within SQL Server and is backed up during SQL Saves.

In the XFT platform we've already addressed this issue and provide as part of the package the solution in source code format

Details to follow…

*Handling **gigantic** file downloads*

Some Trading Partners we know of exchange files in the 35-75 GB range. Far too large to be done using an AWE script since AWE scripts run under the GSB executive as tasks. But Event Rules that are triggered run at about the same level of priority as the GSB executive itself. Instead of using AWE to initiate a file transfer we use an Event Rule. With our solution we can instantiate multiple downloads concurrently and if you're fortunate enough to be running in an Active-Active Failover environment we can cause the processes to run across all of your AA servers.

In the XFT platform we've already addressed this issue and provide as part of the package the solution in source code format

Details to follow…

Recovery and Restart

Every exchange needs to plan for recovery and restart because it is going to happen, a lot at first as you get things going and 'tweak' the system. Plus any 'hotfixes' you need to apply will necessitate a restart of the EFT service. Remember GSB is a Do-It-Yourself kit, The SkunkWorkZ Online has extended the GSB product with a number of AWE scripts (known to us as TPRs) and Event Rule processes to get you up and going quickly.

You need to plan for mid process interruption, failure of the other side to respond, receipt of a mal-formed file and duplicated files to name a few.

In the XFT platform we've already addressed this issue and provide as part of the package the solution in source code format

Details to follow…

Can EFT monitor an email mailbox?

You it can handle POP and IMAP mailboxes and will 'pull apart' the email message such that each component within the email message is given to the process as a separate object.

In the XFT platform we've already addressed this issue and provide as part of the package the solution in source code format

Details can be found on the Downloads page under Read e-mail & Pull Apart

Can EFT send an HTML email?

You can send HTML email messages and imbed application specific data inside the email using the XFT extension.

In the XFT platform we've already addressed this issue and provide as part of the package the solution in source code format

Yes, check out this link Registration

How do you monitor the services for problems?

Monitoring of the EFT service can be difficult in that a number of circumstances external to EFT (such as system SAVEs and SQL Restores) can lock up a service and prevent Event Rules from firing and triggering tasks. Using tools like SCOM don't help because the service didn't fail and throw a detectable error it is simply 'waiting' in a deadlock of some sort for some resource to be released.

To detect if your services are responding we have in development a Heartbeat Monitor that sends a file via sFTP to each node and expects a response file back within a certain time frame or it reports the node as needing to be investigated.

This requires multiple servers and is best initiated from an outside server maybe as a SAAS product.

In the XFT platform we're already addressing this issue and will provide as part of the package the solution in source code format and a HOW-TO subscribe to a SAAS product also.

Details to follow…

How do you monitor for file backlogs?

By structuring the Inbound and Outbound folders by Trading Partner and specific use within we can use a thick client we've built that shows graphically and provides audile alerts when necessary of file backlogs.

In the XFT platform we're already addressing this issue and will provide as part of the package the solution in .NET source code format and a HOW-TO subscribe to a SAAS product also.

Sample can be viewed here.

How do you bulk load hundreds of users?

We use a .CSV file created by the site owner and using the COM API and a VB script program import into the respective FTP site the users with passwords etc.

In the XFT platform we've already addressed this issue and provide as part of the package the solution in source code format

Details to follow…

How do you generate a list of files inbound and outbound without ARM?

We have created a TPR that is started whenever a file arrives at its landing zone area that records to an XML file all the details available about the file and the circumstances of its arrival that are possible. These XML files can be imported into a relational set of tables and analyzed and used for billing purposes or tracing files.

In the XFT platform we've already addressed this issue and provide as part of the package the solution in source code format

Details to follow…

How do you prevent a process from running until some threshold number of files are in the landing zone folder?

We have created a TPR that is started whenever a file arrives at its landing zone area that when triggered counts the number of files in the folder and if it is equal to or exceeds some threshold value it processes the files. These XML files can be imported into a relational set of tables and analyzed for patterns or tracing files or used for billing purposes.

In the XFT platform we've already addressed this issue and provide as part of the package the solution in source code format

Details to follow…

We're an EDI shop, what can you do for EDI users?

We have created a TPR that is started whenever a file arrives at its landing zone area that pulls apart the ISA, GS and ST headers of EDI files, matches the information in them to the contents of an XML file and sends the file along to the filesystem paths designated in the XML file for files matching the criteria. We also have a thick client written in VB.net that graphically shows file counts in folders for monitoring purposes. The EDI file is archived and optionally ZIP'ped/UnZIP'ped and PGP encrypted/decrypted before being handed off to the next step in the workflow. Email notifications to Trading Partners are optionally generated regarding file handling.

Using our Instant sFTP site you can be exchanging files securely with EDI Trading Partners immediately. Check out our FREE web based sFTP process here.

In the XFT platform we've already addressed this issue and provide as part of the package the solution in source code format

Details to follow…

How do you prevent long running processes from triggering more than one instance of a file handler?

We use a GATING mechanism or a Virtual Resource that acts as either an OPEN or SHUT gate. When a long running task begins the first thing we do is look to see if the ‘gate is OPEN or SHUT. If SHUT that means the process is already in motion if OPEN we SHUT the gate and begin the process.

In the XFT platform we’ve already addressed this issue and provide as part of the package the solution in source code format

Details to follow…