Alternative to BizTalk s File Adapter

Source:
http://www.tech−archive.net/Archive/BizTalk/microsoft.public.biztalk.general/2008−01/msg00061.html

- *From*: Jake Watkins <JakeWatkins@xxxxxxxxxxxxxxxxxxxxxxxx>
- *Date*: Mon, 7 Jan 2008 18:03:00 −0800

BizTalk's File adapter is probably one of the most heavily used adapter in BizTalk, or at least in my experience it is. Especially during development. There isn't really an easier way to test out an orchestration than to just hook it up to a file adapter and see what happens. As it happens a lot of integration work deals with processing files. This is where the problem begins to appear.

The File adapter works perfectly for monitoring a single directory for files with a specific file name pattern (or all files if you use *.*). Unfortunately most situations require monitoring a lot of directories. Take the scenario of a convenience store chain that has 1000 stores. The system they've built over the past decades transports flat files between the stores and the head quarters. They have several different file types that get moved back and forth between the store and HQ. In HQ each store has a directory and inside each store's home directory is a directory for each file type the store sends to HQ. Each store also has a server that has directory for each file that gets sent to it. Once the files arrive programs (executables, scripts, or batch programs) work on the files. One file that might be sent to the store would be an item file used by the POS to determine the cost of an item when it gets scanned. The program that works on the item file might take the flat file data and use it to update the store controllers database.

Also to add another little twist that isn't that uncommon: the files are all zipped or perhaps encrypted. That would mean that we would need to unzip and uncompress the files prior to passing them through FFDASM and on in to the message box. You can do that in the pipeline easily enough, but again: it's another head ache.

Using BizTalk in the HQ data center you would have to create a receive location for each directory. Alternatively you could write a custom adapter. You could write an isolated host adapter that takes care of watching the directories, doing the preprocessing and then submission to the BizTalk database. I've run in to this program many times and have some experience with developing it. The issue with this approach is that it always takes longer than expected. It also becomes a headache to change the processing around as business needs change. Finally, it costs money to develop, support, maintain and change.

In order to try and solve this problem (and others) I went off and developed a program that makes working with files as easy as possible. Initially I just called it File Processor, but my friends wisely said that name was too bland. So I renamed it Grinder.

Grinder is a service that will monitor directories for files and then process them using a workflow. It supplies a library of custom workflow activities designed to deal with files. Included are activities like Copy, Move, Execute program, FTP GET, FTP PUT, HTTP POST, HTTP GET, zip and unzip. Because all of the activities are just Windows Workflow activities you can write you own activities. I am also working on expanding the library to include other tools (a BizTalk specific activity would be nice). However, using the included activities it is pretty easy to configure a workflow that will unzip a file and the post it's contents to a BizTalk http receive location (or any web site).

Included is a simplified workflow designer that limits the choices available to focus around processing the file. Using this designer there is no code activity, branching or looping. You can only build a simple sequence workflow. The intended audience for this designer is analysts and system administrators. The expectation is that you are dealing with the file not its contents. If you need to deal with the contents you can write a custom workflow activity for them to use. Alternatively you can write a workflow using Visual Studio and take full advantage of the Workflow runtime. Information on how to access Grinder's services is included.

The reason to use Grinder is because it is a File adapter on steroids. You can easily configure it to monitor thousands of directories, have it preprocess the files prior to delivering them to BizTalk and only have to manage a few receive locations (or just one) inside of BizTalk. It is also cheaper and requires fewer resources so you could deploy it to your file servers without breaking the bank.

Right now I'm giving away a limited number of free licenses because I want feedback. Go here: http://www.engineofprogress.com/Products/Grinder.aspx to read about it and register for a free copy (limit 1 ;). The free copy includes a year of free support which includes updates, enhancements (new activities, etc), and general technical support for the product.

––
jakew@xxxxxxxxxxxxxxxxxxxx

.