INFORMATICA | MARKETPLACE

B2BDT Generic Streamer: Processing Large Files

By: Ophir Chen

Challenge:

When the source file is very large (bigger than 100 MB), one would need to use the Streamer component of DT. The DT streamer is essentially a component that allows DT to process incoming files in chunks by splitting them in different ways (e.g. tokens found in source, structure, iterations, etc.).

With some of the streaming/splitting cases, one might find that the usage of the DT Streamer is not trivial, or insufficient for different reasons. A classic example would be breaking the file into Header-Body-Trailer, sections which is very popular break up with Industry standards (e.g. HIPAA), COBOL and others. That will require more logic than one can apply with the out-of-the-box provided functionality of StreamerMarkers, where the breaking of the source structure is dynamic.

Note:

Please note the terminology of

1. Chunk and buffer. A chunk is a piece of data taken from the entire data. In DT, all pieces of data are normally stored in memory. We refer to those chunks in memory as buffers, but they are essentially the same.
2. Logical buffer vs. Physical buffer. A physical buffer is the raw chunk(s), or part of it, of data that was drawn off of the source. A logical buffer is only the part of the physical buffer which holds a complete section relevant for processing.

Solution-High Level:

The solution is to take advantage of the enhanced scoping, marking and overall text processing capabilities of the DT Parser, and move the parsing aspect of a Streamer into a parser, instead of attempting to handle it via the Streamer.

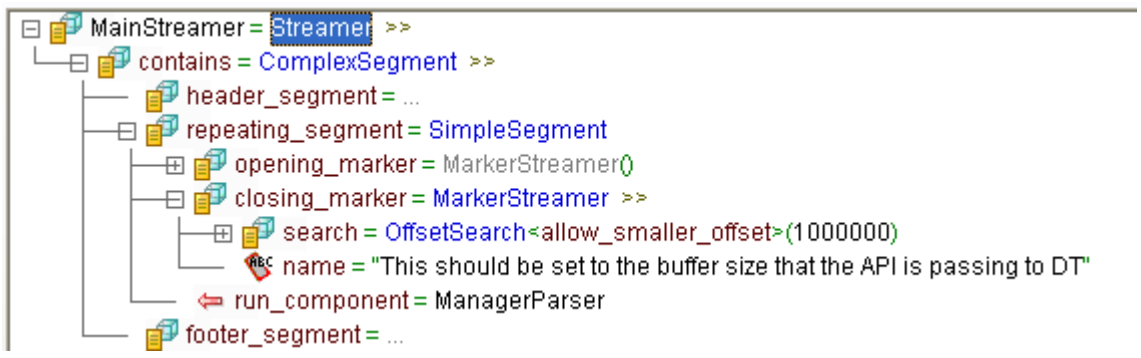
Please note, this is only for those cases where the use of DT Streamer is not a no-brainer. It is still recommended to use only the Streamer if it can easily accomplish the task you have.

Solution-Detailed:

The idea is to treat the source data coming into the Streamer as a pass through buffered, chunked (by size, not a specific logic) data, fed into a Parser. It, in turn, will handle all the logic composing those chunks into logical components by breaking chunks based on the logical components they hold, aggregating chunks if needed and/or appending left over from previous chunk(s).

The below lines describes the components from the attached template project.

1. MainStreamer is the main component in the project.
    a. It's only looking for a user determined buffer size, using the MarkerStreamer.
    b. As long as the file holds buffered data in size that is larger or equal to the offset (in the provided sample 1MB, but you can change it as needed), then that buffer will be what is considered the logical buffer passed to the "ManagerParser".
    c. The last chunk may be smaller than the offset.



2. ManagerParser is the secondary component component, invoked by the Streamer.
    a. It manages the processing of a given buffer and/or appending leftovers.
    b. Appending leftover common scenario is when a logical buffer(s) was processed of the physical buffer. There could potentially have remained some leftover data in the physical buffer after all relevant logical buffer pieces were processed.
    c. The leftovers if exist will be mapped to a designated variable, who's data will in turn, be appended before the new physical buffer (note the use of pre-processing by transformers to add the leftover to the next physical buffer). The variable is reset after its data is appended.

3. Within the "find a logical component to pass to processing component" Group you should implement the logic finding the right scope of data (logical buffer), capture it and then call your parser/serializer/mapper to do the job.
4. The last content is picking the leftover if any exists after the logical buffer was processed out of the physical buffer.