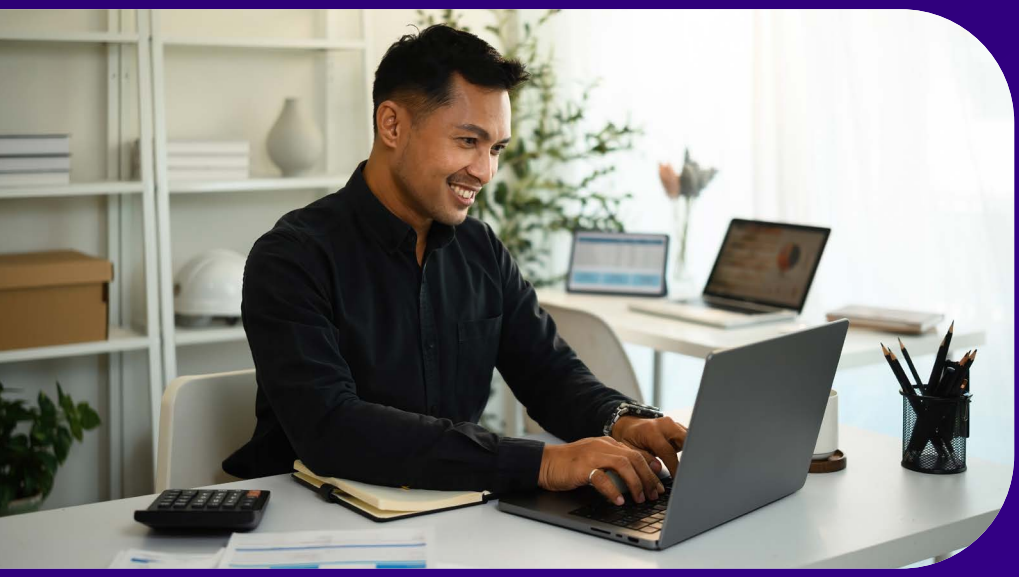
Claria | boomi

Traditional Development vs Boomi + AI



Written by: **Chris Fry** (Claria Senior Sales Engineer) and **Sam Alty** (Boomi Developer)

1. Use Case

The original implementation was designed to perform an extract, transform, load (ETL) execution, which spanned three separate processes and various technologies:

- → MFT scheduler to retrieve a CSV file from an SFTP server and write file to a folder in an AWS S3 bucket for staging purposes
- → Runs an AWS ECS task to retrieve a file from the S3 bucket, performs data mapping to create a new CSV file, then writes newly created file to the same S3 bucket
- → Uses MFT monitor to identify new CSV files within the S3 bucket, creates a copy of files, and writes them to a separate external SFTP server

2. Boomi Manually

The process in Boomi is much simpler as we dont need to move the file to a new location to work on it we can work on it in memory.

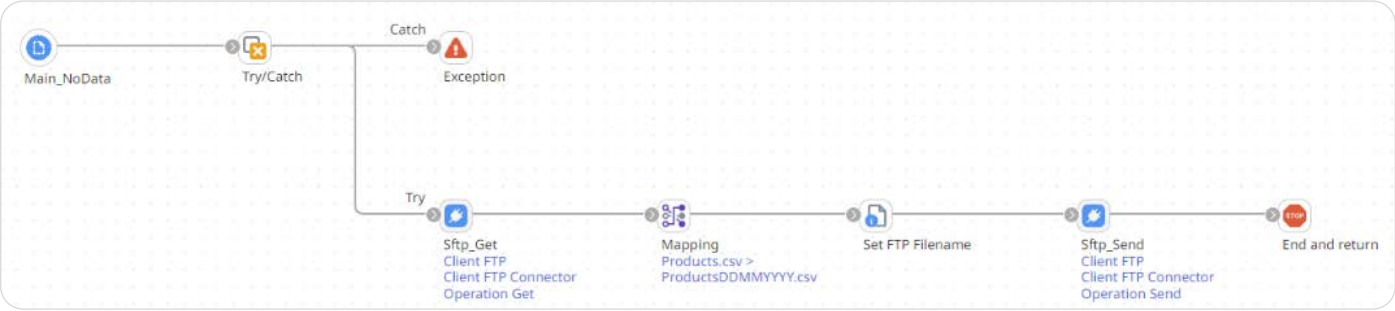
1. Pick up the file from FTP
2. Map the file from one CSV format to the new CSV format
3. Set the new filename and send to the new FTP Location

3. Boomi AI

Using Boomi AI we tried 2 queries one with the mappings included and one without, both produced the same result

1. Create process *PRODUCTS.csv from FTP to ProductsDDMMYYYY.csv on FTP*

2. Create a process which: *Queries source server to retrieve 'PRODUCTS.csv' file via file transfer protocol; Map 'Id', 'productDescription', 'Cost_Price', and 'productHierarchyDescription1' fields from 'PRODUCTS.csv' file to 'description', 'sku', 'price', and 'category' fields in new destination file respectively; Assigns destination file a name that incorporates the current date in YYYY-MM-DD format; Uploads the newly created file to a separate SFTP server.*



4. Results

Traditional Development

- **Estimated development time:** 2 days (16 hours)

Boomi Manually

- **Time spent creating connections:** 9m 54s
- **Manual build time:** 25m 45s
- **Total time:** 35m 39s

Boomi AI

- **Time spent creating connections:** 9m 54s
- **AI build time:** 7m 43s
- **Total time:** 17m 43s

Traditional Dev vs Boomi Manual Time Saving

15h 24m 21s

2693%

Boomi AI vs Boomi Manual Time Saving

18m 5s

50.3%

5. Conclusion

- Boomi AI provided an effective template for our integration requirements
- Significant time efficiencies by expediting the build process
- Little benefit to formulating complex prompts in the hope that Boomi AI will perform more complex builds
- Boomi AI included error handling without being explicitly prompted to do so

Get in touch

Do you want to improve your systems?

Ask our consultants without compromise. We will help you find the best solution for your project.

Contact us