MAVIDIAN TECHNOLOGIES

perfection matters

DATA CONVEYER™

INTRODUCTION

Imagine that a weather forecast announces a temperature drop below 20 degrees. Does this indicate the arrival of an arctic air mass? In America, most likely yes. In Europe however, where temperature is measured in Celsius, this weather forecast would simply convey a relief from a heat wave.

Data is everywhere, but it is not useful unless we know how to interpret the data provided to us. In the ever-evolving technology universe, we've become accustomed to conveniences that help us in interpreting data. For example, a **SAVE AS** feature:

- ✓ There is a Word document. Let's **SAVE** it **AS** a PDF file.
- ✓ There is a jpeg image. Let's **SAVE** it **AS** a bitmap.

In a corporate world, things tend to get complicated. The **SAVE AS** feature cannot be easily applied. There are just too many proprietary standards and "one-of-a-kind" conventions.

Wouldn't it be nice though, to take the **SAVE AS** feature for granted?

- ? There is an X12 file with EDI 837 transactions representing medical claims. Let's **SAVE** it **AS** a keyword file, so that we can load claim transactions to our database.
- ? There is text data to be loaded to our system. Let's **SAVE** it **AS** records in our database table.
- ? There are periodic transactions received by our Enterprise Service Bus. Let's collect them and **SAVE AS** hourly Excel spreadsheets.

With Data Conveyer, these statements are not far from reality. No, Data Conveyer does not come with a **SAVE AS** button to click, however it reduces the time and effort to implement data migrations, often by orders of magnitude. Hence, a corporate data-centric project can be expected to take days or weeks to complete, not months or years.

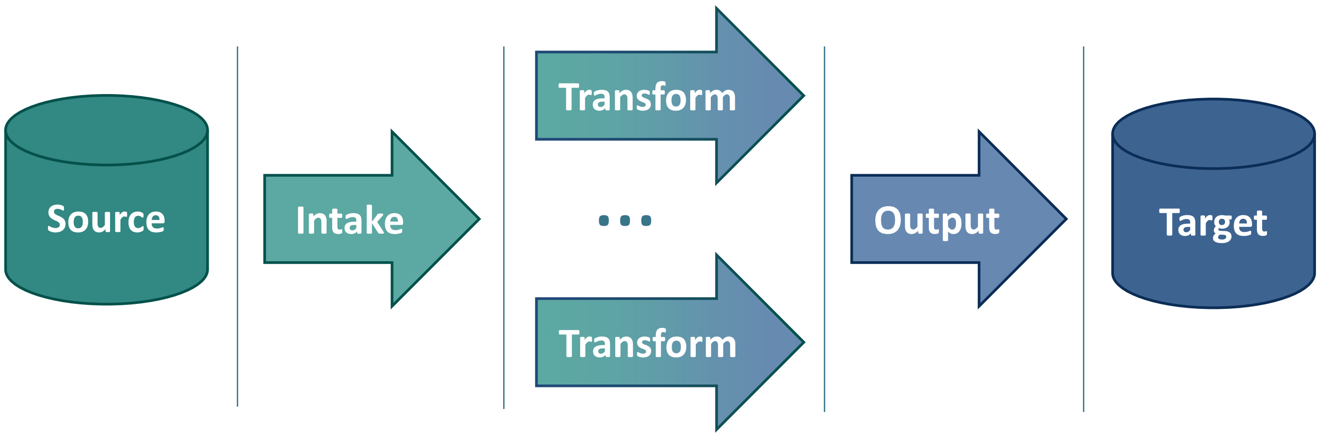
And... if you're creative enough, there is nothing preventing you from creating your own **SAVE AS** button that employs Data Conveyer to address your particular scenario.

DATA CONVEYER™

OVERVIEW

Data Conveyer is a toolkit and a lightweight transformation engine to facilitate real-time data migrations. It harnesses the power of modern, multi-core processors by making extensive use of parallel processing and concurrency. The toolkit enables rapid implementation of a variety of transformations and integration patterns.

Data Conveyer's architecture conforms to the Extract-Transformation-Load (ETL) paradigm.



MAIN CHARACTERISTICS OF DATA CONVEYER

- A self-contained executable (DLL library) that does not require any dedicated infrastructure for deployment. It can be deployed on any environment running .NET Framework 4.5 or .NET Standard 2.0. This includes Windows, Linux, Mac OS and other systems.
- Non-blocking, asynchronous execution mode. Operation in progress is cancellable.
- Inherently parallelizable, scalable and production ready.
- Highly configurable, comprehensive, yet intuitive API that accelerates prototyping, development and implementation of various data migration scenarios.
- Native support for a diverse set of data formats, including delimited values, fixed width fields, key-value pairs (keywords), XML, JSON, and X12 (EDI).
- Expandable architecture to support additional data formats and transformation patterns, either natively or using plug-in modules.

DATA CONVEYER™

ADVANTAGES FOR SOLUTION DEVELOPMENT TEAMS

Feature	Description
Elimination of mundane, yet time-consuming tasks	Little or no effort is required to configure Data Conveyer to load data, parse input rows, combine related records, convert data formats, produce output records and perform many other tasks common in data migration projects.
Separation of concerns	Tasks to be performed by solution developers are simplified because Data Conveyer isolates their contexts. As an example, in order to implement data mappings, a developer only needs to supply a single function that translates a set of uniform input elements into output elements.
Ease of workload distribution	Components of solutions based on Data Conveyer are loosely coupled. Accordingly, tasks performed by team members working on a single solution do not generally depend on each other. As an illustration, work on receiving/extracting data from a custom data source can be conducted independently from implementing data mappings.
Accelerated prototyping	Data Conveyer defines default values for all configuration settings. Furthermore, components applicable to common scenarios are built-in and do not need to be supplied. For example, if a data mapping function is not supplied, Data Conveyer will simply pass unchanged contents of input records onto output.
Improved testability	Solutions based on Data Conveyer naturally fit into the test-driven development (TDD) paradigm. Data Conveyer can be leveraged to quickly build tests that supply arbitrary input records and expect corresponding records on output.
Agile and DevOps friendly	Data Conveyer facilitates software delivery in small increments. Working Software can be constructed in a single Sprint cycle. Each subsequent cycle builds upon the previous one by gradually introducing new functionality.

EFFORT REDUCTION

A typical data migration project of moderate complexity involves a development effort of roughly 300 hours when using traditional techniques. The table on the right compares such estimate to the approach based on Data Conveyer. While the actual savings depend on the project specifics, a 50% effort reduction can be reasonably expected and the rate can be as high as 90% in case of proofs of concept or pilot projects.

Approach / **Task**	Traditional	Based on Data Conveyer
Requirements Analysis	20	20
Solution Design	20	10
Coding - Data Handling	100	0
Coding - Business Logic	100	90
Quality Assurance	50	20
Deployment Support	10	10
Total Hours:	**300**	**150**

DATA CONVEYER™

DESIGN GOALS

UNIVERSAL FUNCTIONALITY The idea behind Data Conveyer stemmed from involvement in numerous data-centric projects and realization that they generally follow the same pattern differing only in some "variable elements". These elements, once identified, became configurable settings to be integrated during orchestration of a uniform data migration process. To support such flexibility, Data Conveyer features a powerful API (Application Programming Interface) that consists of over a 100 of configuration settings and dozens of user-definable functions.

EASE OF USE To ease complexities of its API, Data Conveyer employs a comprehensive logic to assume default values and actions whenever configuration settings, functions or parameters are not supplied. In fact, the entire Data Conveyer process can be orchestrated and executed in a single-line statement. While more code is required in non-trivial cases, the settings and types exposed by Data Conveyer are designed to be intuitive and naturally fall into place.

POWER Since introduction of functional programming features into .NET (such as LINQ, Task Parallel Library, Dataflow, and many others), it became possible to succinctly write powerful code implementing functionality that was previously only attainable via complex, multi-tier software. This power comes at a price of a relatively steep learning curve. Data Conveyer aims at leveraging all power of modern .NET features while interacting with its caller using a familiar object-oriented paradigm.

EFFICIENCY Related to the previous item is the overall efficiency of Data Conveyer based solutions. In contrast to traditional approaches, there is no need for dedicated infrastructure, such as database or application servers. Instead, solutions run in memory relying heavily on parallelism, concurrency and asynchronous processing. To minimize memory footprint, data is subjected to rigorous buffering as it passes through Data Conveyer.

EXPANDABILITY In its early stages, Data Conveyer has undergone major refactoring episodes, which in fact included a complete rewrite at one point. The main motivation behind these efforts was desire to facilitate future product expansions. The current architecture of Data Conveyer makes it straightforward to introduce support for new data formats, new types of transformation as well as other features, for example logging.

EASE OF DEPLOYMENT Data Conveyer is a standalone executable. As such, it gets deployed by a simple file copy, regardless if for development or production purposes. There are no dependencies to load (aside from .NET), no Registry updates, no installers or uninstallers to run. Data Conveyer can also be deployed as an industry standard NuGet package.

DATA CONVEYER™

FUNCTIONALITY AT A GLANCE

GENERAL

- Built-in support for various data formats including delimited values, fixed width fields, key-value pairs (keywords), XML, JSON, and X12 data (HIPAA transactions).
- Support for strongly typed data elements.
- Data buffering throughout the process to contain memory usage.
- Cancellable while in-progress.
- Progress reporting.

TRANSFORMATION

- Parallel processing (multiple engines).
- Support for various transformation modes, including per record and per cluster.
- Flexible API allowing different ways to access and manipulate data to suit circumstances and individual preferences.

INTAKE

- Connectivity to any data source, real-time or offline, in either synchronous or asynchronous mode.
- Support for multiple sources.
- Variety of settings to fine tune data parsing process.
- Ability to combine related records into so-called clusters.

OUTPUT

- Connectivity to any target, real-time or offline, in either synchronous or asynchronous mode.
- Support for multiple targets including custom routing rules.
- Variety of settings to fine tune data formatting process.

PLANNED FOR FUTURE RELEASES

- Build-in support for additional data formats, such as HL7 data.
- Pipelined (cascaded) transformations.
- Additional transformer types.
  - ✓ Aggregator
  - ✓ Sorter

QUALITY ASSURANCE

From the outset, Data Conveyer has been subjected to stringent quality assurance measures. Every build of the product involves a series of over 400 test cases containing a total of more than 6,000 assertions.

DATA CONVEYER™

WHEN IS IT USEFUL?

Data Conveyer can be invaluable in a variety of data migration solutions, where it can act as:

- Data Importer/Exporter
- Uploader/Downloader
- Format Converter
- Data Adapter
- Data Cleanser/Scrubber
- Data Analyzer
- Data Auditor
- Transaction Recorder
- Pre-processor
- Post-processor
- Rules Engine
- Transformation Engine
- Transaction Router
- Transaction Splitter
- … and more …

Listed below are sample scenarios illustrating value added by Data Conveyer in terms of manual effort reduction, acceleration of delivery timeframe, or less tangible benefits, such as quality improvement and risk mitigation.

SCENARIO ONE: RAPID IMPLEMENTATION OF NEW BUSINESS

An insurance company signs a new deal with a 3rd party vendor. The deal necessitates data exchange between the enterprises, such as import/export of membership data, claim data, and the like.

Data Conveyer allows rapid implementation of a production-ready solution to translate data between vendor specific and internal formats. Specifically, such implementation becomes a matter of days or weeks, as opposed to months when using traditional techniques. Furthermore, a proof of concept prototype of inbound or outbound data exchange can be provided in a matter of hours, which may play an important role in project planning.

SCENARIO TWO: INTEGRATION INTO EXISTING INFRASTRUCTURE

In order to import external data, a database system relies on batch processing of keyword data files. However, inbound transactions are received in real-time by a company-wide enterprise service bus.

In this scenario, Data Conveyer can be employed as a listener to cumulate real-time transactions received from the bus, translate data to the required format and post the resulting files to the respective batch jobs.

DATA CONVEYER™

SCENARIO THREE: AUTOMATION OF BENEFITS CONFIGURATION

A team of product consultants works on configuring new products to be offered during the next benefit year. The process is tedious and consist of repetitive cycles. Consultants define configuration data using ad-hoc Excel files. The files get uploaded into the database in order to conduct user acceptance tests. At the conclusion of the tests, the cycle needs to be repeated, until the test results are satisfactory.

In this scenario, Data Conveyer can be leveraged to generate SQL statements from Excel data, which offers two-fold improvement: reduction of manual effort and error elimination. As an added benefit, the operation can generally be conducted directly from the consultants' desktops.

SCENARIO FOUR: USER FRIENDLY DATA RETRIEVAL

An application receives complex data from a remote source, such as a web service. Before applying the application logic, the data needs to be parsed, scrubbed and/or evaluated. In addition, the application's graphical user interface needs to remain responsive during this process.

In this scenario, Data Conveyer can do both: consume the service as well as extract required elements from received data… all in a non-blocking, asynchronous mode.

SCENARIO FIVE: EASY SEPARATION OF EDI TRANSACTIONS

EDI translators often identify inbound transactions based solely on a transaction type. As an example, EDI 834 transactions may represent daily enrollment updates as well as monthly summary records. Consequently, it may be difficult to separate transactions intended for distinct designations.

Data Conveyer can be used to evaluate transactions produced by the EDI translator, apply custom routing rules and create separate outputs to forward transactions to distinct targets for further processing.

DATA CONVEYER™

SCENARIO SIX: DATA ANALYSIS AUGMENTATION

During a system conversion, data profiling is performed to assess data quality and compatibility with the target system. Very large quantities of data get loaded into a SQL database for subsequent analyses. It is quickly discovered that majority of the records contain no data of interest. In addition, several data fields are universally empty. It is not difficult to bypass such irrelevant data using SQL queries; however, the sheer volume of data heavily affects performance and makes the analysis process very time consuming.

Data Conveyer can easily eliminate loading unneeded data in the first place. Records containing no relevant data can be filtered out and empty fields can be removed, both before data gets loaded into the database.

SCENARIO SEVEN: ENHANCED EDI TRANSLATION

EDI transactions received from a newly signed trading partner cause occasional errors when processed by the EDI Translator. Upon analysis, it was determined that some transactions, while HIPAA compliant, are interpreted as duplicates instead of being recognized as distinct transactions. Any reconfiguration of the EDI Translator is problematic, due to the effort involved and also the high risk of impact on already established business.

Data Conveyer can be employed in this scenario as a preprocessor to evaluate and update EDI transactions according to custom rules, so that they are disambiguated and interpreted by the EDI Translator as intended.