**EDAM X12 EDI 834**

An Open-Source Initiative

Last Update: 2023-05-29 – Added details on Instances Processing and other minor updates.

EDAM (“Enterprise Data Asset Management”) Studio is an open-source project originally sponsored by Datovy. The initial conception of EDAM was in the form of a command-line console application that could take a schema (XSD, JSON Schema, and others) to be used to flatten into an EDAM Studio type specification. Given an X12 EDI transaction set specification such as those that follow, EDAM Studio could be used to derive other representations including (XSD, JSON Schema and others):

* 274 – Health Care Provider Information
* 820 – Payment Order / Remittance Advice
* **834 – Benefit Enrollment and Maintenance**
* 837 – Electronics Claims

# EDI Workbook

EDAM Studio uses an X12 EDI specification (Excel – OpenXML based) workbook that contains the following columns:

|  |  |
| --- | --- |
| SegmentName | Segment Name |
| Entity | Entity (such as a database schema) |
| EntityName | Entity Name (that could represent a database table) |
| EntityElementName | Entity Element Name (ako a table column) |
| EntityLink | Entity Link (relate the entity with others that may include Foreign Key). |
| Position | Establish the position of the TAG Element |
| SegmentCode | Segment TAG Code (such as “ISA”, “NM1” or other) |
| SegmentRepeat | Occurrence of TAG (“0:1” (optional), “1:1” (required), “1:N” (unlimited)) |
| SegmentRequiredType | Such as “R” = Required, or “S” = Situational |
| SegmentReference | Segment Element identifier such as “ISA01”, “NM101” or other |
| ElementID | Element ID |
| ElementType | EDAM Studio Element Type such as “element”, “code” or other) |
| Element | Element Name |
| ElementDescription | Element Description |
| ElementRequiredType | Element Required Type (“M” mandatory, “O” optional) |
| DataType | Data Type such as “string”, “date”, “number” or other |
| MinimumLength | Minimum Length |
| MaxinumLength | Maximum Length |
| Loop | Loop Identification |
| Parent | Parent Loop Identification |
| Codes | The segment element expected codes (if any). |

The “Entity” columns will be used to help in deriving an alter JSON or database representation of instances. Those in “brown” are critical for the proper generation of derived document structure. Use the “Codes” column to add expected codes such as qualifiers, entity code ids, or other such as NM1 entity-code-id or other applicable to the EDI document.

# EDAM Studio EDI Project

To illustrate the use of the “EDI Workbook” by the EDAM Studio the “Datovy.EDI” project is provided in the “Edam.App.Data/Projects” folder (look it-up in the “Files” folder), there the “834” sample workbook will be found. The project will load the specification into an EDAM Studio flat representation of its document structure as illustrated in the provided screenshot and could be used to generate mapping specifications, data-dictionary, XSD, JSON Schema, and others that may provide a starting point for derive artifacts as those may be needed on particular situations. The screenshot shows the hierarchical structure, in this case, of the 834 document.

While selecting the generation of XSD’s EDAM Studio prepare 2 files, the first with the EDI common elements such as (complex) types that will include Elements like “ISA”, “GS”, “N1” and others as defined in the Workbook, and a second file with the particular message at hand and in this sample scenario particular to the “834” transaction set.

# X12 EDI Document Instance Processing

As of this writing Datovy is wrapping up additional work to allow the validation and processing of X12 EDI document instances initially based on XSD’s as generated by EDAM Studio.

## EDI 834 Code Sets

TAGs such as NM1 that may need an “Entity Code ID” based on a code-set has purposely kept out of the Workbook but all of those will be provided separately as “pipe” (“|”) delimited csv files. Understand that all the codes within a code-set may not apply to a particular transaction set and for this Datovy do have loop-up code resources for supported EDI documents that in conjunction with the csv files will be provided in a different GITHUB project.

## Document Instance Processing

On this release the processing of document instances has been added and their testing started. The “Edam.B2b” project contains needed classes to support EDI processing including:

* EdiDocument – Base class that provides persistence of loaded definitions as represented by the EdiSegmentInfo class.
* EdiDocumentReader – Given an EDAM asset data element list created following the definition file (see EDI Workbook) all entries are used to prepare the EdiSegmentList that is a collection of segments sequenced based on expected loops with the necessary information to search for particular segments using loop, parent loop ids, or qualifiers (such entity-codes) and follow through the loop set to help the generation of segment instances Instances (see EdiInstance).
* EdiInstance – Helper class that will have the collection of instance documents. Each instance starts with an ISA segment and will contain all identified segments while reading (in this case) an 834 document. While processing an instance segment loop may be repeated per its definition and each repeated item will be added in the order they are found.
* EdiInstanceReader – Class methods to read instance documents by calling the “FromFile” method that parses the file and generate corresponding Instance segments.
* EdiSegmentInfo – Define supported segment details including loop and parent loop details.
* EdiSegmentList – Collection of segment details.

## From Instance to JSON

Once an instance document has been read, parsed, and all instance segments have been added to an EdiInstance segment (set) list (each representing individual ISA sets found with the given file) then the “Entity” columns as defined in the EDI Workbook are used to create a JSON document with all the data found in the instance file fully mapped to the entity.

Note that the Entity columns may be set to target a (kind of) database schema, tables and columns. Alternatively, is up to the EDI Workbook Architect, BA or developer to set their preferred language to be used to map instances targeting a JSON document. It is possible that one or more Workbooks be offered to target different use cases.

Feeding the JSON back within EDAM into their own project data-dictionaries, JSON-LD, further mappings and use-case definitions and other artifacts may be generated.

# Final Thoughts

Hopefully other specifications will be added including 837, 274, and 820 (in that order) that provide the open-source community an alternative option to other existing commercial offerings.

GITHUB Open-Source Projects

You can access these projects through these links:

EDAM - <https://github.com/esobrino/Datovy.Edam> (EDAM Studio)

HC-DST - <https://github.com/esobrino/Datovy> (Health Care – Disease Surveillance)

Contribute to EDAM

If there is any interest in learning about EDAM and/or contributing to the projects, follow the above links and let us know. EDI 834 instance documents to test the solution, further testing of EDAM functionality, refactoring code, adding missing components and other useful efforts will help to improve this offering.

Hope to hear your thoughts soon,

Eduardo

Graphical user interface, text, application

Description automatically generated with medium confidence

(EDI 834 Document - Screenshot)