Etasoft 

Etasoft – XTranslator Lite 1.x

EDI XTranslator Lite designed for automated and semi-automated translation of EDI X12 files to and from CSV or XML. It supports all EDI X12 40xx, 50xx, 60xx, 70xx transaction types.

Setup

For Windows install the software using provided setup file xtlitesetup.exe.
For Linux install the software unzipping the file xtlitelinux.zip.

Windows installation contains both GUI and command line translation tools. Linux package only contains command line translation tool.

Initial Steps

What will you need to have translations setup:

1. Documentation describing EDI segments in PDF or other format. Documentation should list all the segments in the EDI X12 message.
2. Sample EDI file(s) you want to process.

Both requirements are important. You might get away without sample EDI files if documentation describing EDI segments is very detailed. It is also possible to mock-up EDI files by creating them manually in text editor and then use them as sample EDI files.

Ask your EDI trading partner to provide documentation and sample EDI files for you. Once you have it follow these steps:

1. Load new transaction layout using “New”.
2. Add segments based on your documentation. Set qualifiers for added segments using Constant value.
3. Set loops by clicking check box next to the segment.
4. Map fields using check boxes next to the elements.
5. Use “Generate Example” to generate sample CSV or XML file if your output is EDI.
6. Run and test mapping with “Run”.

Overview

Few important terms to get started using XTranslator Lite: map, loop, segment and element.

1. **Map** is a set of translation rules and configuration details stored in a file. Map is used to run the translation. Once you have the map setup then you can run it repeatedly to convert input files into output. Map is usually setup once and runs for years.

2. **Loop** is a set of segments inside EDI transaction. In CSV file it is single line with data fields.

3. **Segment** is EDI segment such as ISA, GS, ST, etc.

4. **Element** is EDI element or sub-element. Translator does not display sub-elements differently. During processing sub-element separator is used to read or write sub-elements. This is handled by default inside translator.

Most of mapping work is about setting up loops, adding copies of existing segments with different qualifier values and test running the map.

Once you load EDI transaction, look for major loops in your EDI documentation, such as detail and sub-detail information. Then select EDI segment that starts the loop and click on its check box, and give loop meaningful name.

On CSV side each EDI loop will be represented by a separate line with fields. On XML side it will be a tag that contains other tags.

Typical EDI segments that start loops are CLM, LX, HL, SVC, etc. EDI transactions that contain HL segments form hierarchical structure. HL mapping requires extra efforts. If you have to produce EDI files with HL segments, note that each HL has to contain unique IDs that point to other HLs in parent-child relationships. Also each HL contains unique ID on they own. Provide IDs inside your CSV or XML file that is used to produce EDI file.

Loop is repeating set of fields in your EDI file. Good examples of loops are lines in the invoice or claim service lines. Consider that all data in EDI files is in the loops. Even if header and trailer information may repeat only once but it will be assigned to belong to specific Header or Trailer loop.

Typical EDI files contain few major loops. Each loop contains data in a set of segments. General names for the loops: Header, Detail, Sub-detail, Trailer. Translation results in multi-level CSV or XML file. Each level corresponds to the loop (group of EDI segments in EDI or group of fields in CSV).

For example: EDI X12 837 Health Care Claim file may contain these groups: Header, Claim, Secondary Information, Service Lines, Trailer. You can give names to loops. If names such as Header, Detail and Trailer may sound too general use more specific names that make sense to you.

CSV file layout may look like this:

```
Header,SenderID,ReceiverID,Transaction Control Number,...
Detail,Claim Number,Patient First Name,Patient Last Name,Billing Provider,...
Subdetail,Service Date,Procedure Code,...
Subdetail,Service Date,Procedure Code,...
Subdetail,Service Date,Procedure Code,...
Detail,Claim Number,Patient First Name,Patient Last Name,Billing Provider,...
Subdetail,Service Date,Procedure Code,...
Subdetail,Service Date,Procedure Code,...
Subdetail,Service Date,Procedure Code,...
Detail,Claim Number,Patient First Name,Patient Last Name,Billing Provider,...
Subdetail,Service Date,Procedure Code,...
Subdetail,Service Date,Procedure Code,...
Subdetail,Service Date,Procedure Code,...
Trailer...
```

Note each line starts with specific text. In this setup “Subdetail” repeats a lot while there is only one instance of Header and Trailer lines in the file.

Same basic layout works for both EDI to CSV/XML and for CSV to EDI translations. But EDI to CSV/XML translations have few additional features: lookup and spread fields.

Steps

XTranslator Lite is simpler to use because mapping consists only of three major steps:

1. marking loop start segments
2. adding new segments with different qualifiers
3. mapping fields using check boxes.

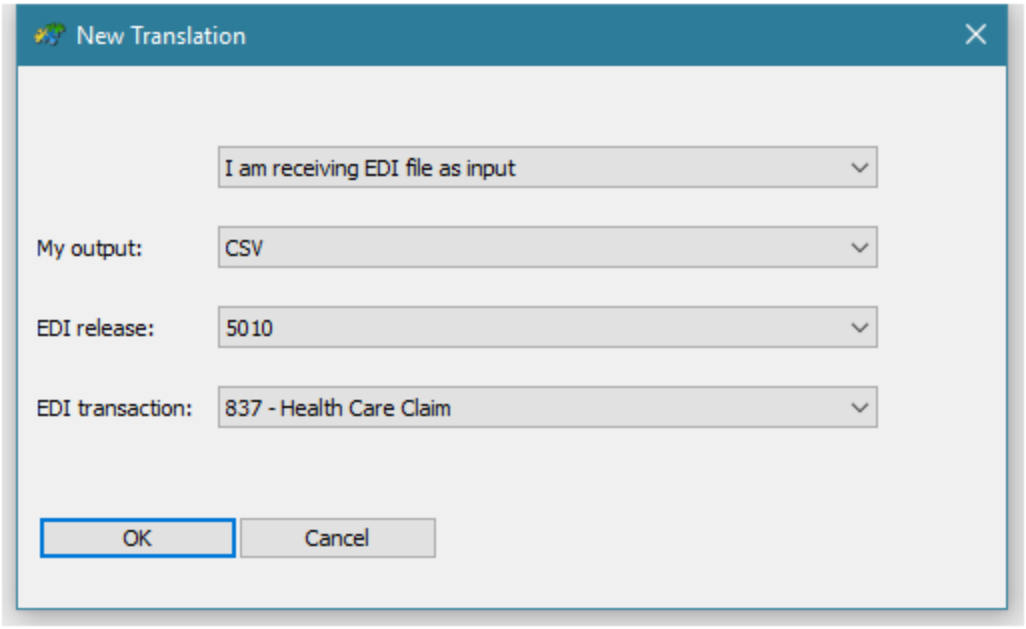
Map

You can setup translation rules and store them in a mapping file with extension “.xmap”. Map lists all of EDI segments for selected transaction, mapped fields, constant values and even functions for dates and control numbers.

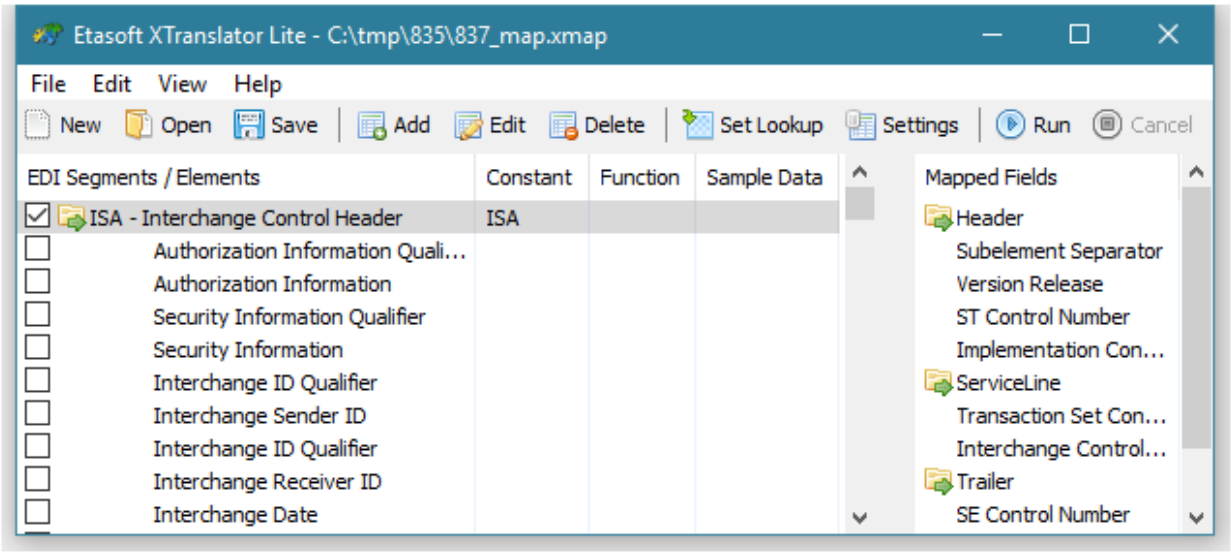
While in the process of setting up the map it is always a good idea to run map whenever few changes are made. This will act as confirmation that you are on the right track. It is especially easy if your input is EDI file and you have a sample EDI file. It is harder if your output is EDI file, then mapping changes may require adjustments to CSV or XML input file. Still try to keep map in “run-able” condition at all times.

When you run map **Sample Data** column contains actual data for that element. Only first instance of data is displayed. If element does not have data or it is blank then translator picks non-blank data for that element from the next loop.

Sample Data is here to guide you. Do not use it for business analysis because blank values are ignored values might be from different loops.



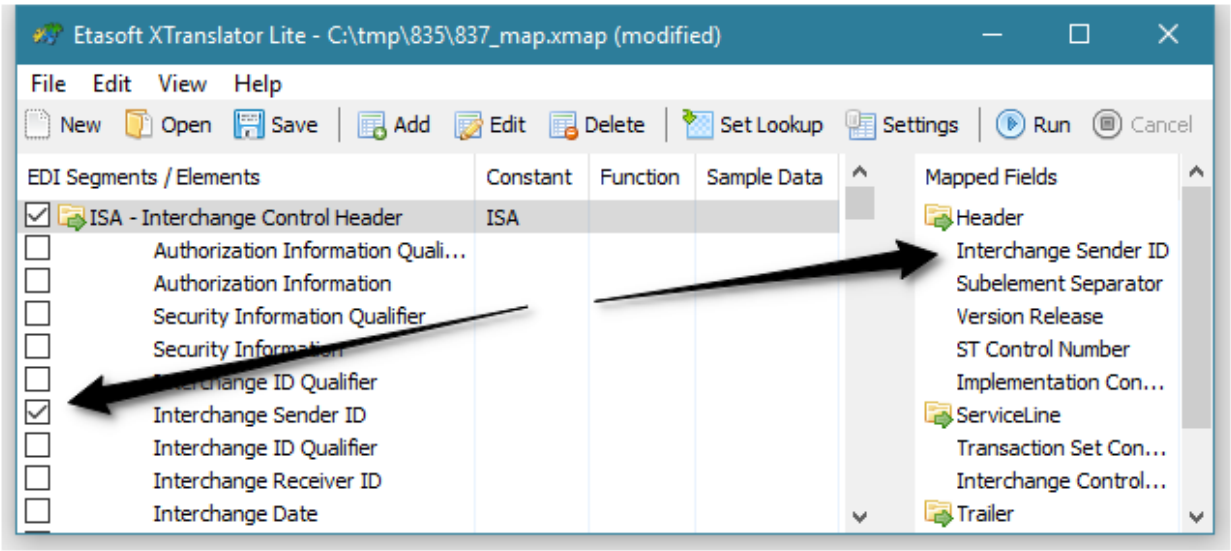
Click on New in Map Editor GUI and choose your EDI release number and transaction type. Once transaction layout is loaded in the Map Editor modify it for your own needs.



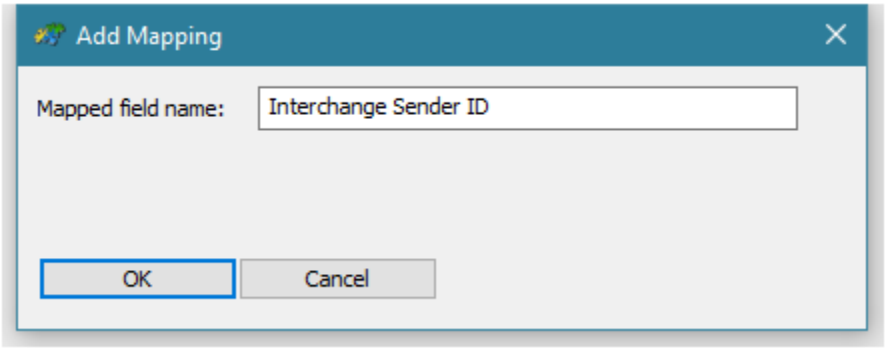
Loaded EDI transaction displayed on the left side of the map with checked boxes next to mapped fields. Right side contains only mapped fields. Adjust the map for your own needs using left side. Right side will reflect changes you make on the left.

EDI segments start with two or three uppercase letters and have **Constant** value set. Elements are displayed slightly indented to the right.

Built-in standard transactions come with Header, Trailer loops and limited number of mapped fields. For typical translation setup to be useful you will need to add more loops and map more fields.



Mapping is simple. Add new mapped field by checking box next to EDI field. Field will be assigned to the nearest loop above it.



Edit mapped field name. CSV files do not use field names for processing. Actual name serves as documentation only. But for XML files field names are important and get produced as XML tags.

XML tags cannot have spaces and certain characters. If XML tag is invalid translator will slightly alter name when producing XML tag. It will remove spaces from tag names and may substitute few other symbols in order to keep XML valid.

Adding Segments

Most EDI segments have general form. For example segment N1 contains name information followed by N3 and N4 segments containing address of that name. In order to understand what that name means it is important to look at the N1 qualifier element(s).

Same applies to many other pieces of information in the EDI file. Qualifiers describe what specific segment or group of segments means. Use Constant column to set qualifiers.

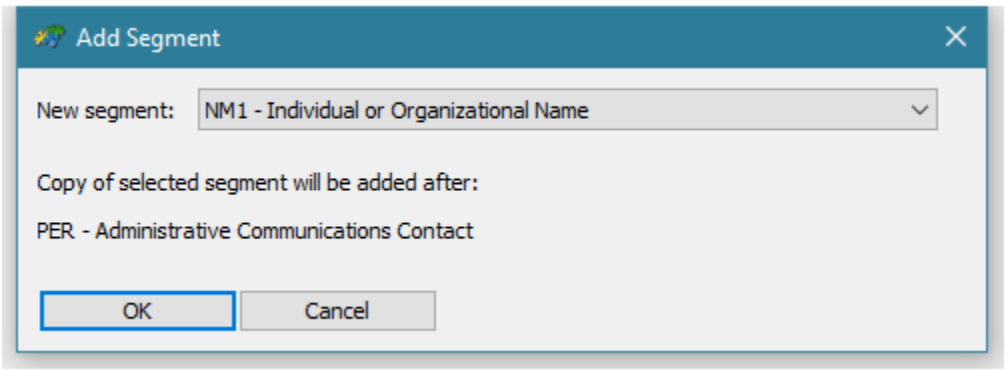
For example: NM1 name information without qualifier value can represent any name. But NM1 with qualifier value “IL” represents a Subscriber Name.

When you import standard template all segments are in general form, and do not have qualifiers. Your EDI file may have qualifiers based on trading partner requirements.

You can set qualifiers in **Constant** column in the Map Editor left view. Consult your EDI documentation and look for qualifier or identifier values. Qualifiers are usually present in first, second or third element for that segment. That makes them easier to spot.

When input file is EDI then translator reads segments and matches them against segment ID set in the **Constant** column. If you have qualifier values set in **Constant** then translator will try to match both segment ID and all qualifier values for that segment.

Usually few segments are next to each other and differ only in qualifier values. If you do not set qualifier values for such of the segments in **Constant** then translator may match other segment with the same name.

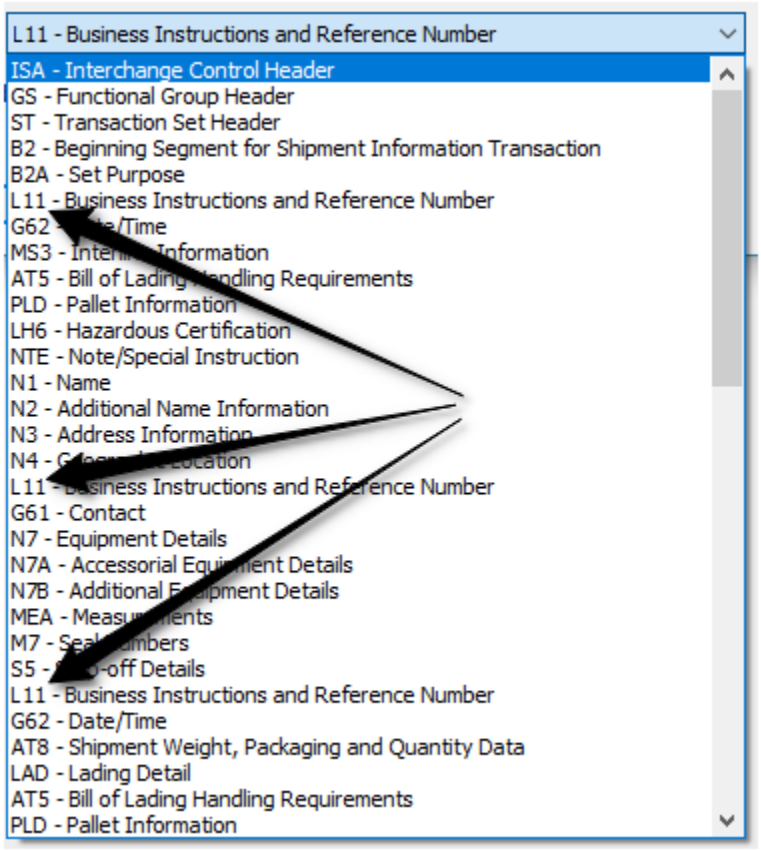


Use “Add” button and add more of the same segments just after current segment. Then set specific qualifier on each segment.

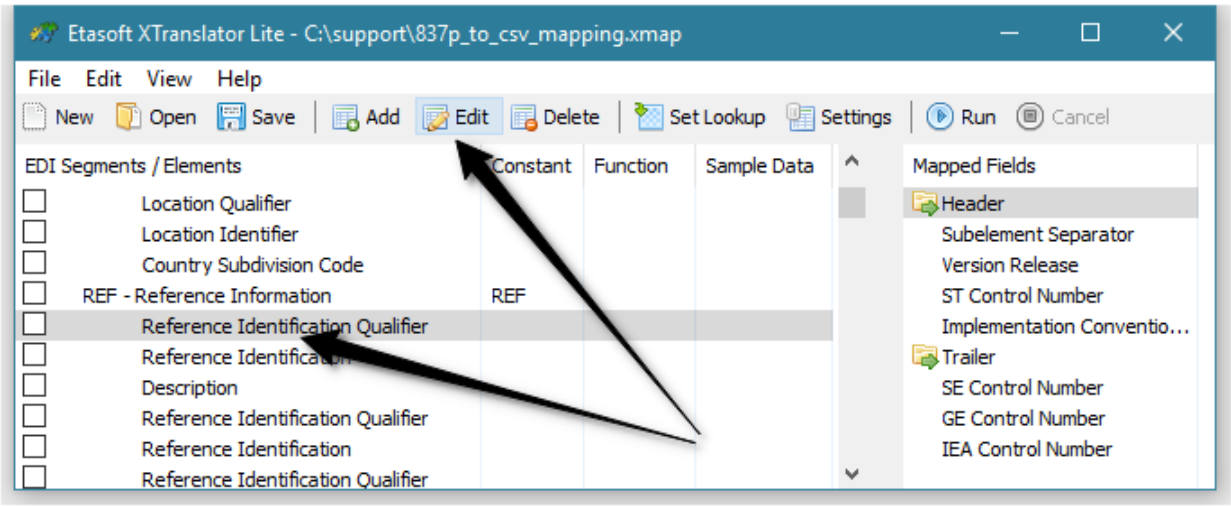
Your modified map segments with qualifier values should match your EDI documentation for that transaction format. For example if your EDI documentation lists two N1 segments with different qualifiers then your map should have two N1 segments each with specific qualifier.

Order for the segments is usually important. So if you have NM1 with qualifier “40” and NM1 with qualifier “41” it is best to have them in the same order as your documentation lists them. Having segments in the wrong order may cause issues during translation when incorrect match will assign data to the wrong segment.

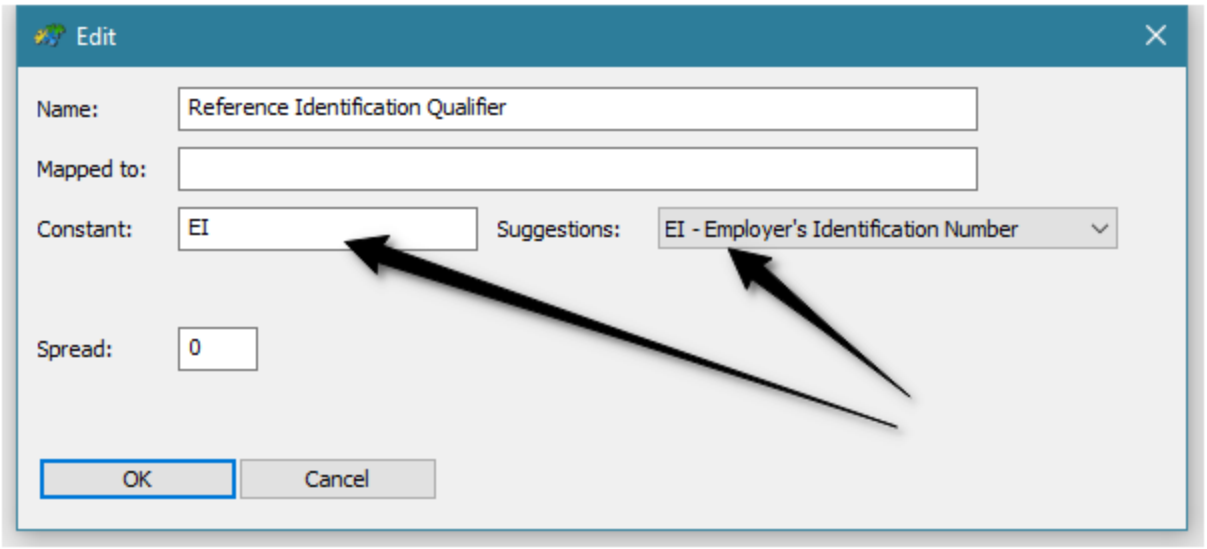
Order is especially important if you have few instances of the same segment located at different points inside EDI transaction. Please see next picture with “L11” segment repeated 3 times in this EDI X12 transaction loaded from standards.



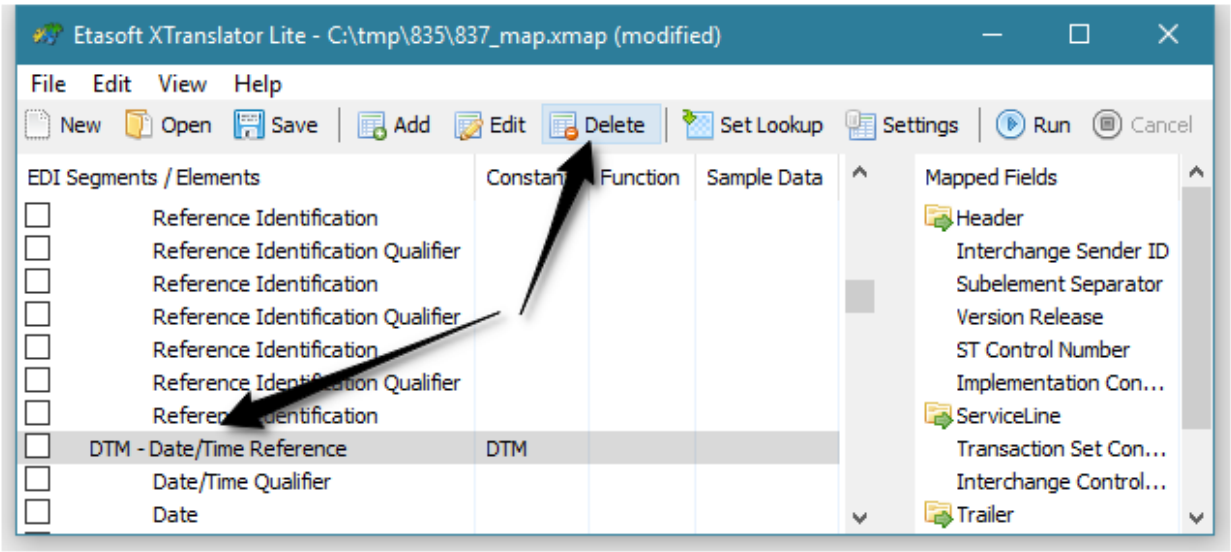
In this example EDI standard transaction has number of L11 segments at different loop levels. Your layout in the map should match what is listed in your EDI documentation.



Once you add new segment then click on the element that has a qualifier, and set qualifier value with Edit button. Double click on the list of elements also opens Edit screen.



If you select specific element all qualifier values for it get listed in "Suggestions" drop down list. You can also manually enter value inside **Constant** and modify mapped field name.

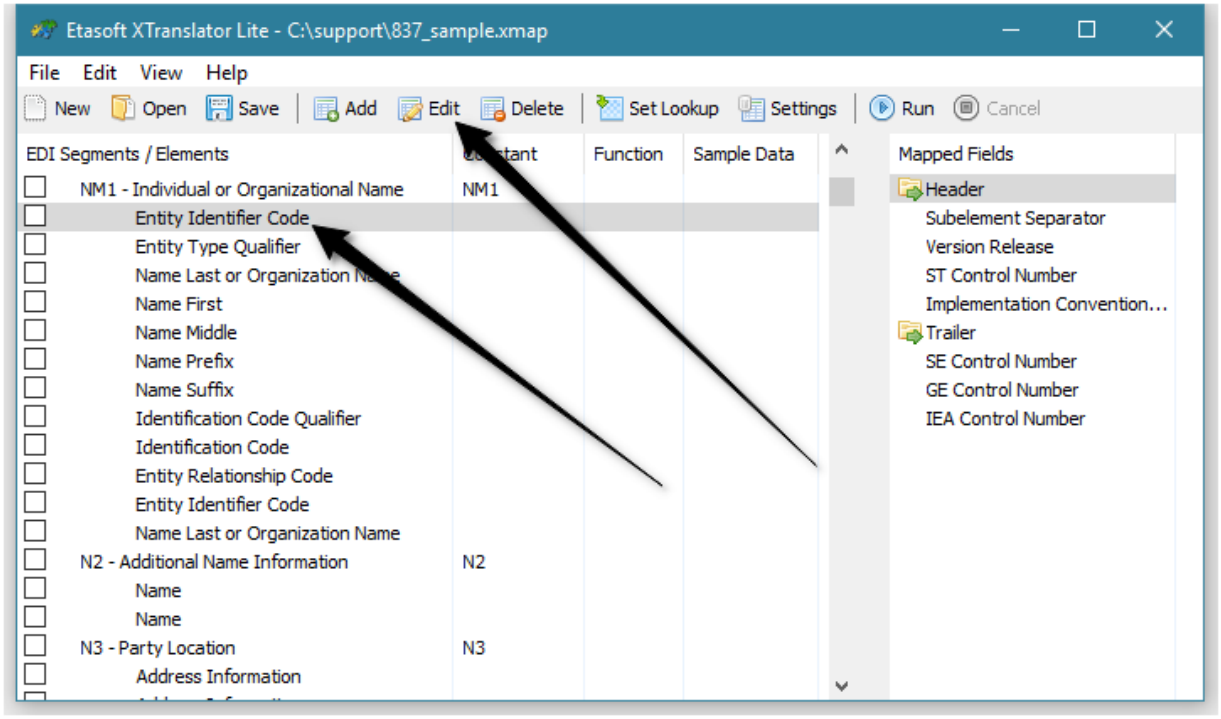


Use Delete button to remove unused segments. You can also delete individual elements if your elements for selected segment are not standard.

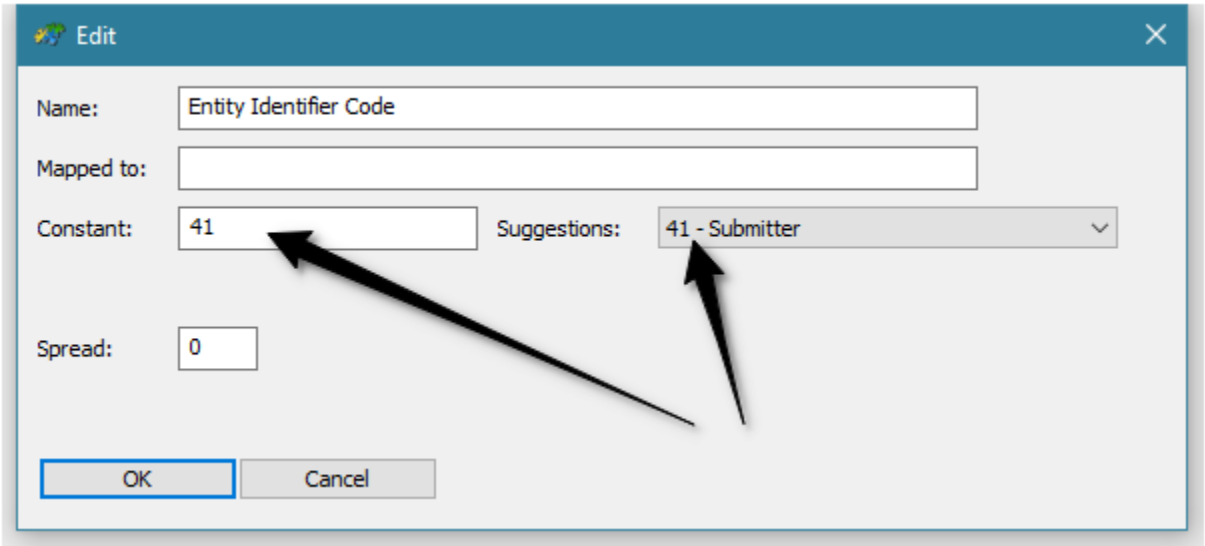
NM1 Setup Example

NM1 segment is popular segment in many EDI transactions. In most cases NM1 is not a stand-alone segment but start of a small group of segments: N2, N3, N4, REF and PER. It is a combination of fields with name, address, identification and contact information.

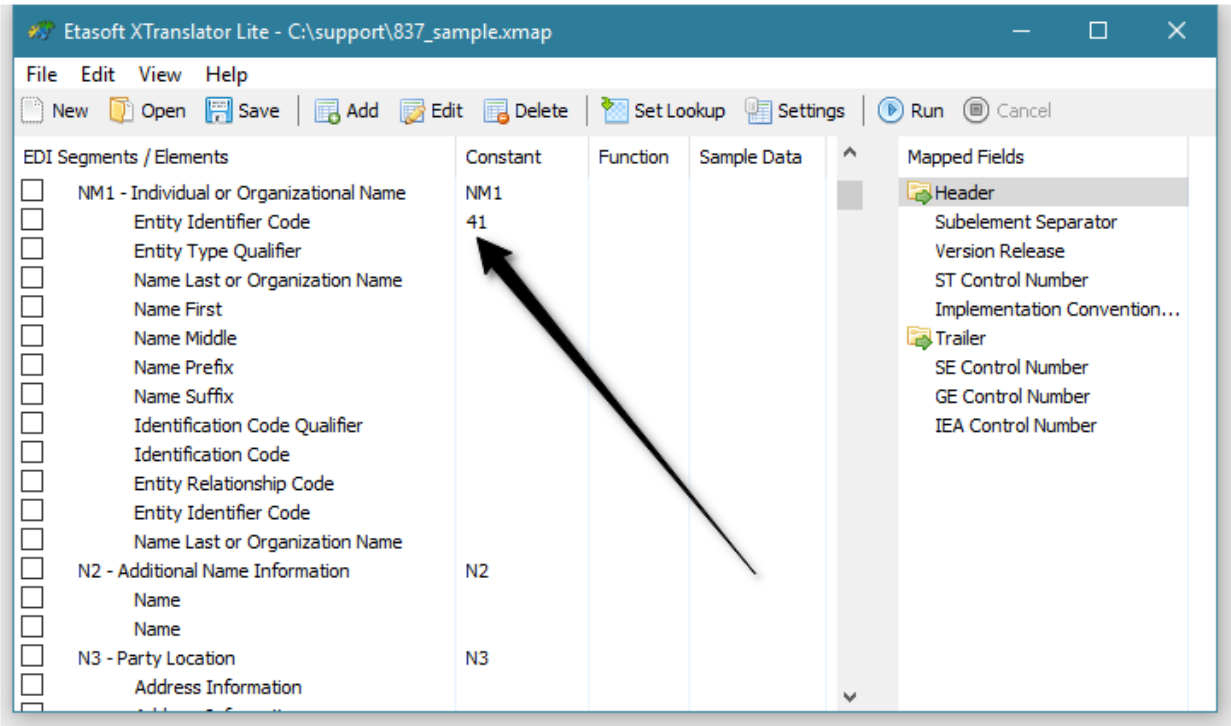
In this example we modify EDI input map to have two NM1s with different qualifiers “40” and “41”. After this modification each NM1 can be mapped and would have separate fields in resulting CSV file.



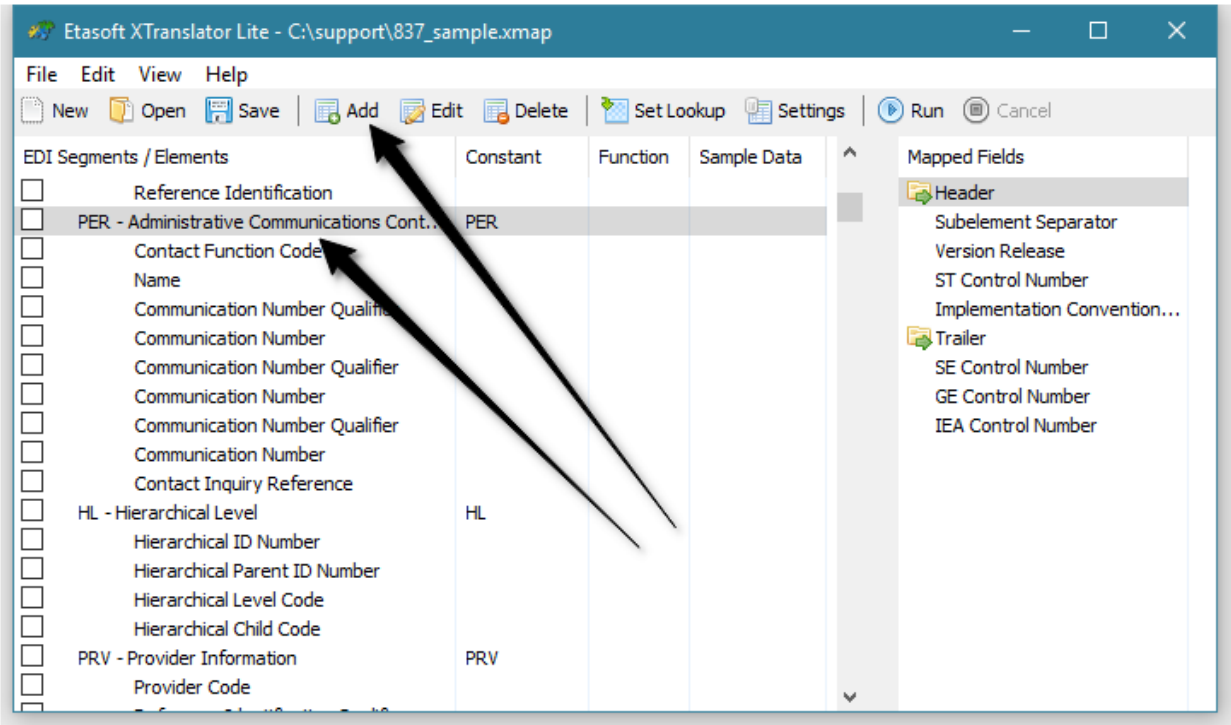
Select NM1 element #1. Since EDI 837 has more than one NM1 we double check and use first NM1 on the list. Then click “Edit”.



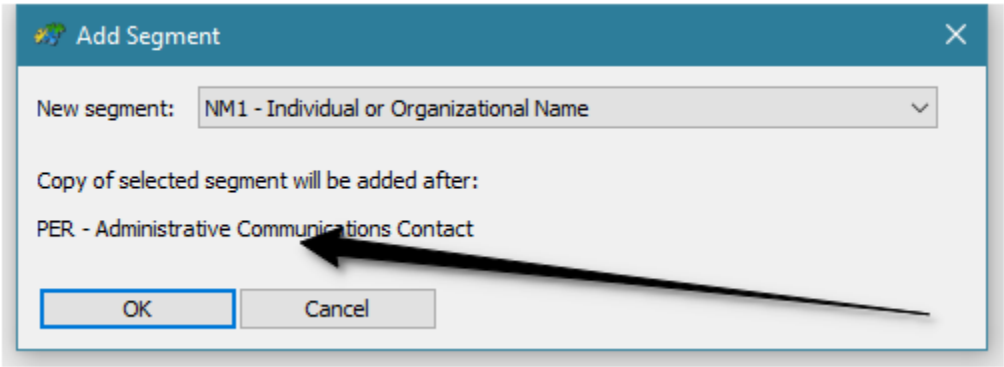
Pick qualifier value that matches our EDI documentation from trading partner.



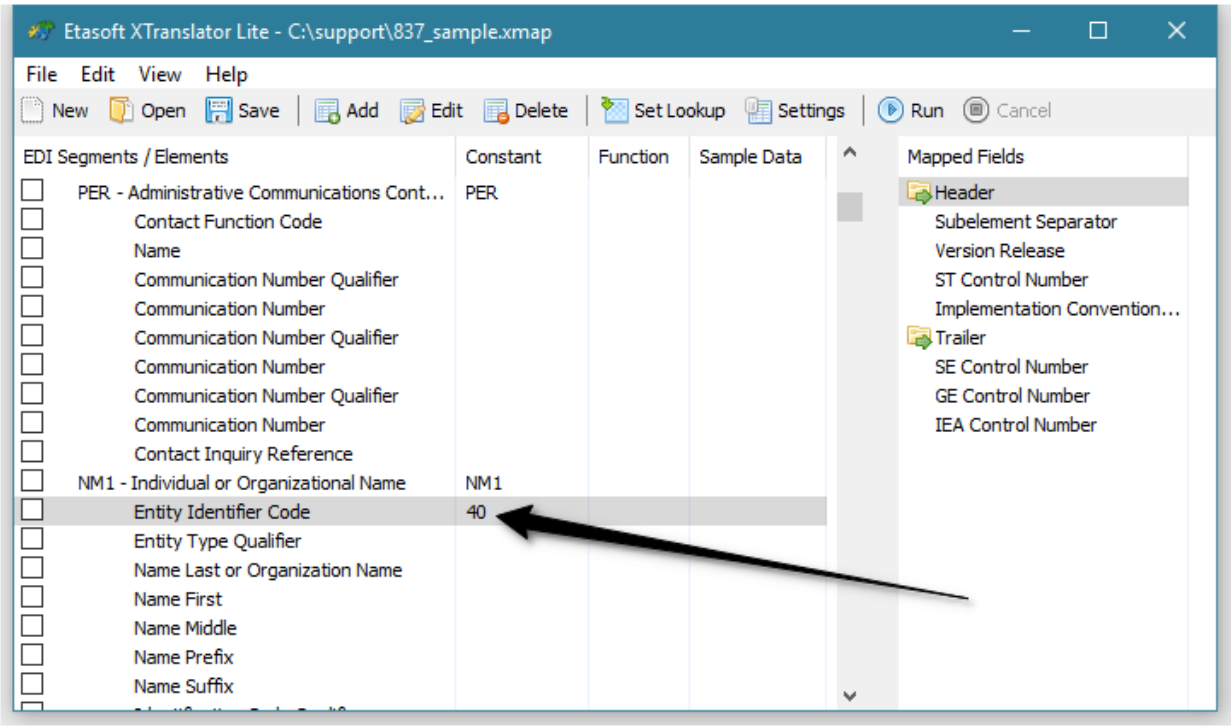
Map Editor picture shows new “41” value in Constant column. Now translator will match both NM1 and value “41” inside element #1 when processing EDI input file.



Since NM1 has group of segments, next NM1 should start after PER.



We add another NM1 after PER.



Set NM1 element #1 to “40” using “Edit”. At this point we are done with these two NM1s and move on to setup other segments in EDI X12 837.

Loops

If you are converting EDI into CSV translator will produce EDI loops as separate lines in CSV file.
If you are creating EDI files then your CSV should have each loop as separate line in the CSV file.

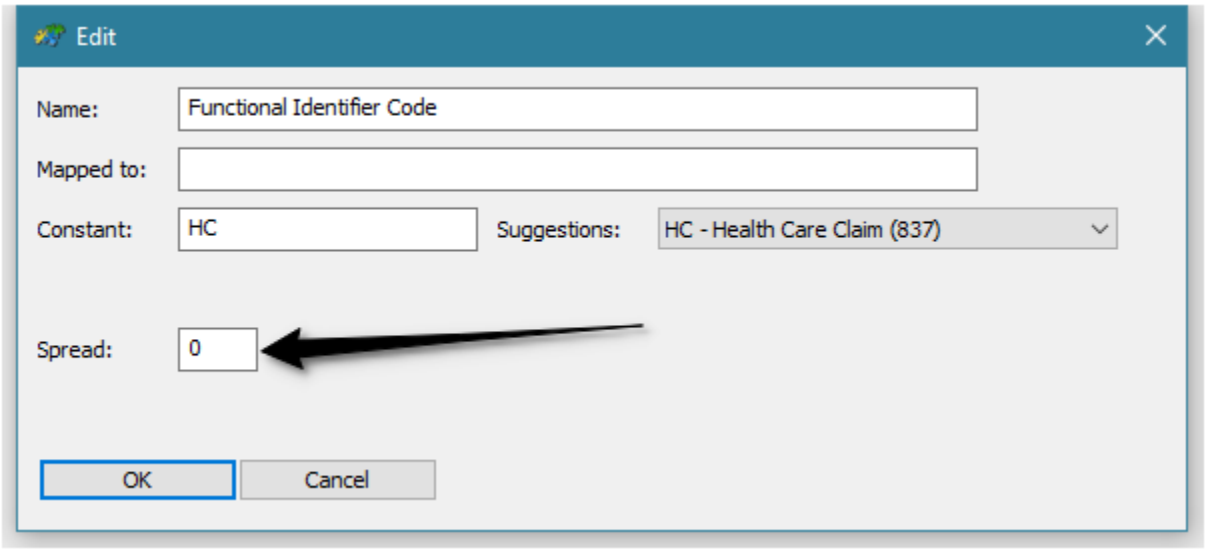
You can start new loop by clicking on the check box next to the segment. Then set loop with unique name. If you are converting CSV into EDI it is the text that will start the line.

You can set it so each line starts with unique text such as Header, Detail or Trailer.

Spread

Use **Spread** for spreading looping data horizontally. It is good option if you have an EDI loop that can only repeat fixed number of times.

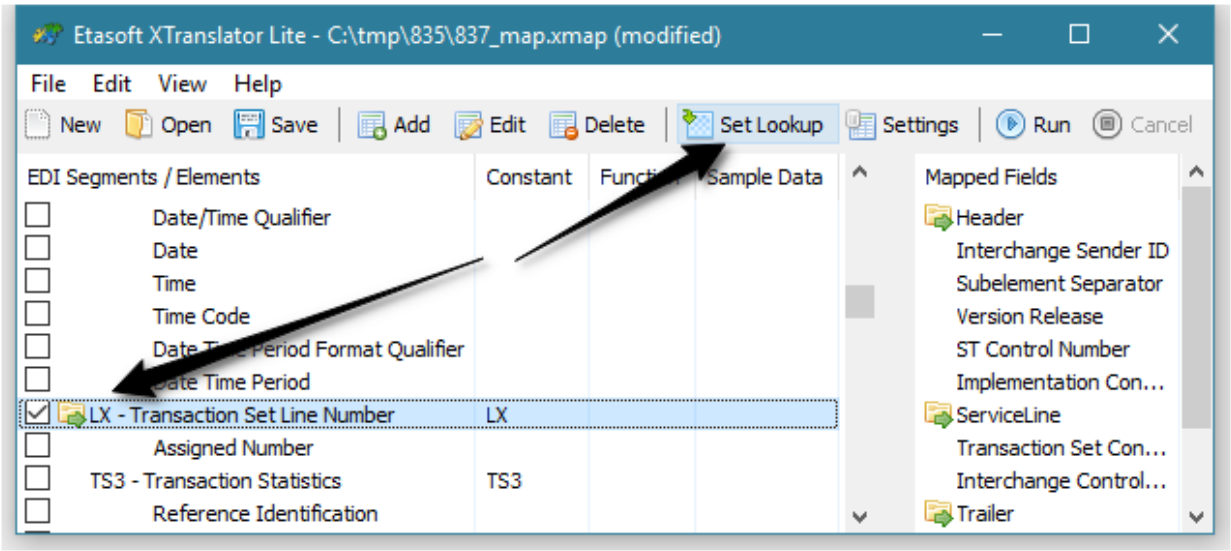
Let say you have a loop that can repeat up to 5 times, you can set Spread = 5 and all loop values will be spread in 5 output fields placed horizontally. This way you do not have to create additional loop (additional line in output CSV).



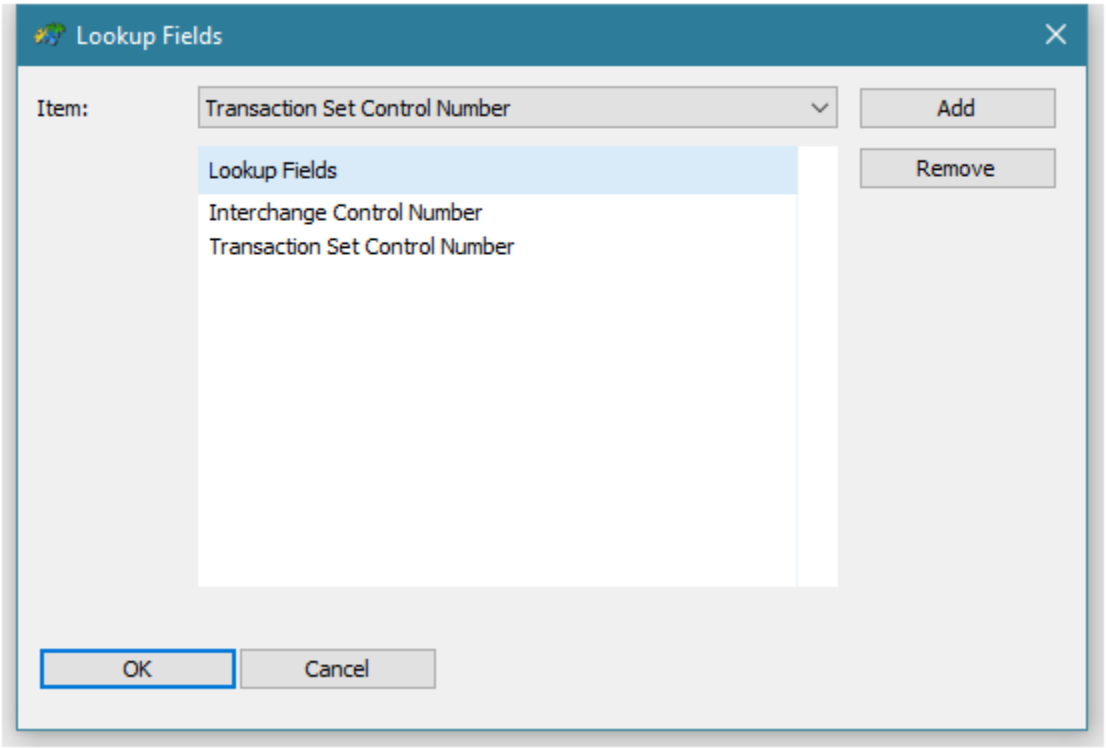
Values get spread from left to right and fill allocated slots (spread fields). If **Spread** is set to zero that means values are mapped to this field and not spread horizontally.

Lookup Fields

Lookup fields bring value from prior loops into the current loop. This helps tie CSV lines if they get loaded into relational databases as separate records. For example Detail loop may need transaction control number or other identifier from the Header loop.
“Lookup Fields” option is available when converting EDI files into CSV or XML.



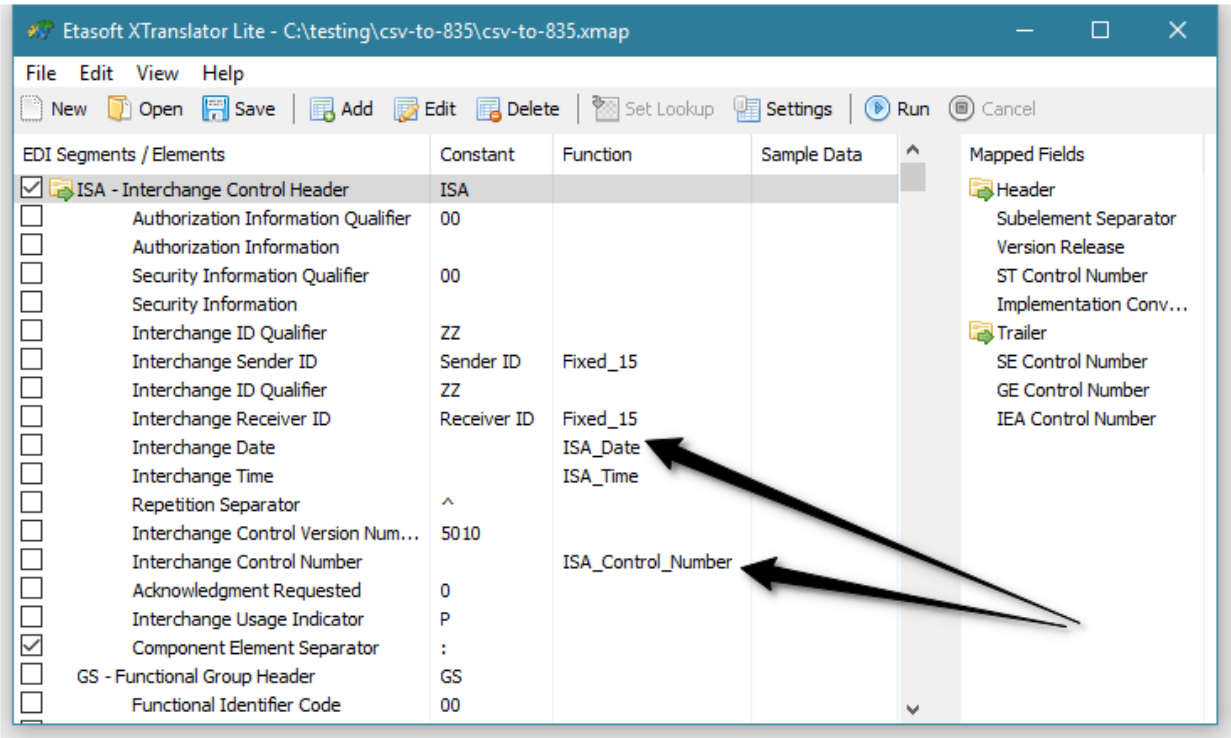
You can add lookup fields only on map items that start the loop.



Add and remove lookup fields using this screen.

Producing EDI

Producing EDI files from CSV or XML has few important differences. Your CSV has to supply all of the dynamic values to form valid EDI file. Translator comes with templates set to help you produce valid EDI envelope segments ISA, GS, ST, SE, GE, IEA. Rest of the fields have to come from CSV or XML source.



Functions help generate current date, time and control numbers for outgoing EDI file.

EDI puts special strict requirements on envelope segments. If improperly formatted files will fail even basic validation. Most fields in EDI are not fixed length. ISA segment is an exception. All of its fields are fixed length. Even single off-by-one character error in any ISA fields will likely to result in processing error.

EDI dates have few different formats such as YYYYMMDD or YYYYMMDD-YYYYMMDD (here “YYYY” stands for year, “MM” for month and “DD” for day).

Few segments contain counts, totals or IDs. Example: LX, HL, CTT. Translator does not provide math functions. Counts, totals and other derived values should be included in CSV or XML input file.

Translator trims unused or blank values from the ends of EDI segments just before producing them in the output file. If input CSV or XML file contains blank value(s) that are at the end of EDI segment then translator will trim redundant blank elements. This is important feature since EDI standards do not allow blank redundant elements at the ends of EDI segments.

Translator provides functions to generate EDI control numbers. EDI envelope segments contain control numbers to track and acknowledge EDI messages. Functions that generate control numbers have suffix "_Control_Number". Control numbers do not have to be strictly sequential but they cannot repeat.

Translator saves control numbers in the files with "controlnum" prefix. File is stored in the same processing folder as your map file. If you want to reset control numbers to initial value of "1" then simply stop the translator and delete the file with "controlnum" prefix. When translator runs next time it will create the file and start counting from "1".

Many segments have control number as last element in the segment. You need to have that element mapped and provide "0" (zero) value for it. Translator trims unused elements from the end of EDI segment. If zero is not provided or is blank, then control number would be trimmed.

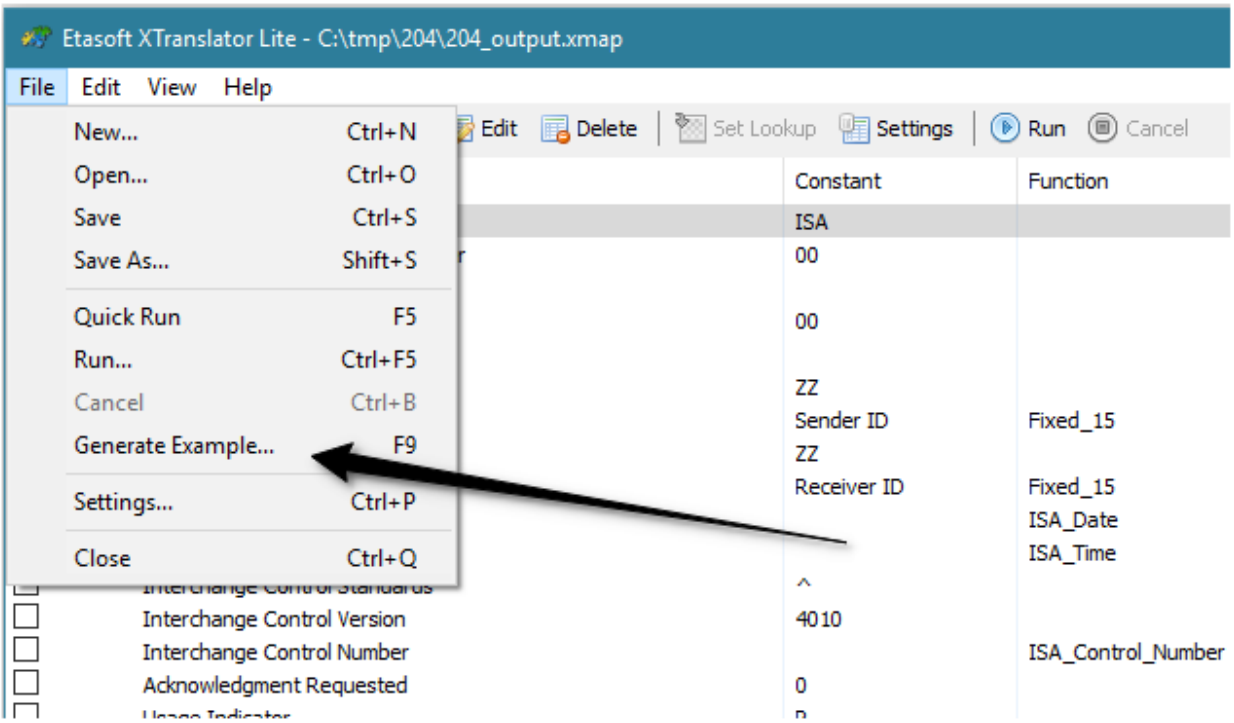
```
Header,:,5010,0,5010
Trailer,0,0,0
```

This picture shows initial basic CSV to get you started. It only contains Header and Trailer lines and fields to produce EDI envelope segments.

Use EDI validation tools to ensure that your produced EDI file is correct. We provide EDI validation tools.

Generate Example

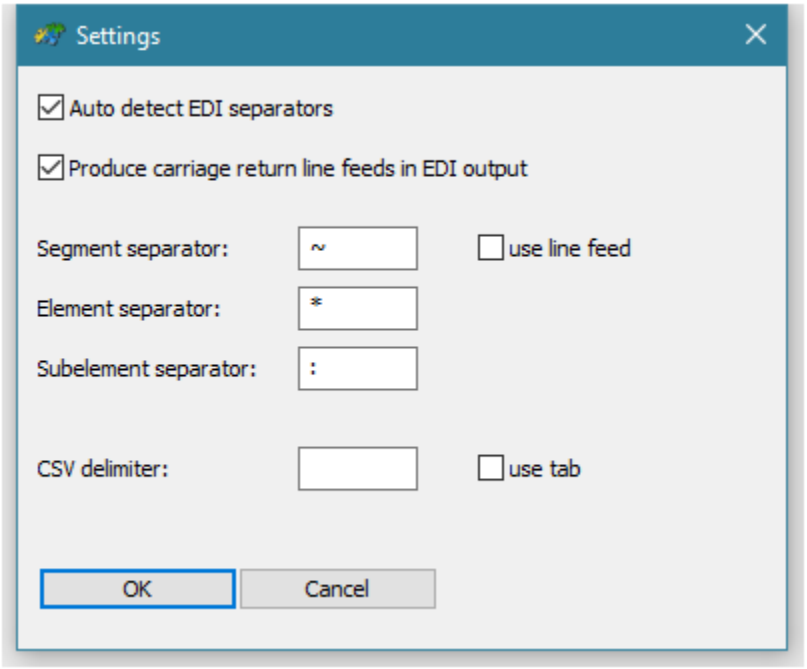
If your output is EDI then it is helpful to have sample CSV or XML with all mapped fields listed. Use File-Generate Example menu to produce sample input file, then place your test data into the generated example and run the map.



“Generate Example” creates example input file with all mapped fields listed.

Settings

You can setup EDI and CSV specific separators using Settings screen.



EDI files may have different separators (delimiters). Set auto-detect of EDI separators or use fixed separators.

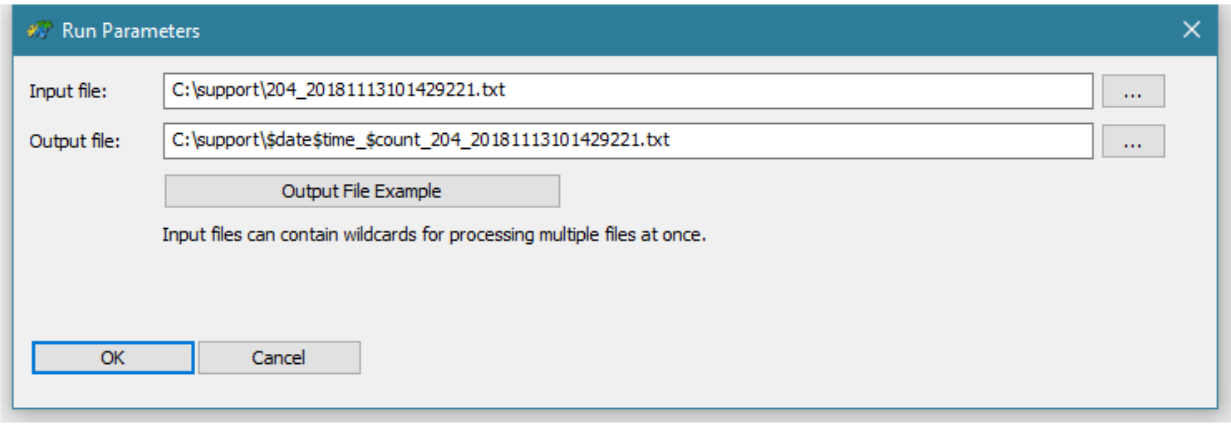
CSV files use comma as field separator. Number of EDI fields may have commas. Example: address fields and company names. When comma is inside the field value double quotes are placed around the value so it would not get treated as two separate fields. This makes CSV processing more complex.

You can avoid complex processing using "|" (vertical pipe) or tab characters as CSV field separator.

Run

Map editor can process single file or multiple files.
If you want to process multiple files then input has to have wildcard and output has to be set with macros to generate dynamic output file names.

Use "Run" and "Cancel" buttons to run and cancel running translation. Run translation requires both input and output files. Simply pick input file name using "…" and press "Output File Example" button. Output file name will be generated with macros.



Macros place dynamic data into output file name, such as current date, time or count number.

List of all macros you can use in output file name:

Macro	Data
$date	Current date in a YYYYMMDD format
$time	Time in hhmmss format
$nano	Nanoseconds
$count	When multiple output files are produced this inserts new count number
$uid	Unique ID. Long hexadecimal string that is guaranteed to be unique with high probability
$inputfile	Input file name will get placed as part of output file name

Input file name may contain star wildcards. For example path "C:\support\837\input*.txt" will process all the input files in the "input" directory with extension "*.txt". Use star wildcards if you want to process multiple input files at once.

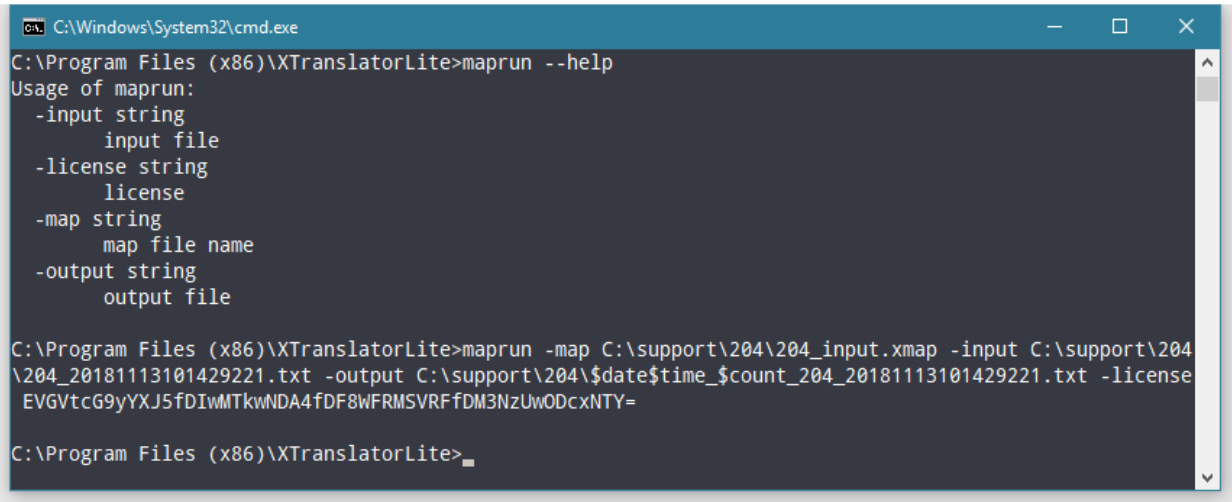
Use “File-Quick Run” menu via F5 keyboard shortcut to execute same translation with most recent “Input file” and “Output file”.

Tips and Tricks

Good directory structure can go a long way in helping streamline processing. Create separate folder for each translation and place subfolders “input”, “output” and map file inside that folder.
Now input and output files will go into they own dedicated folders, and each translation will contain only files related to it.

Command Line Execution

Use utility maprun to run translations via command line. You can run command line tool with “—help” parameter to find all available parameters.



Command line tool parameters:

Parameter	Use
-input	Input file name or * wildcard if processing multiple input files
-output	Output file name with macros. Please look at “Run” chapter above for list of available macros.
-license	License key. Use license key from Map Editor Help-License screen
--help	Print help on screen

If any of the files fail translation command line tool returns non-zero exit code.
Zero exit code means no-error.

If any path to input, output or map files contains spaces then it has to be enclosed in double quotes. Otherwise processing will fail. This is command line specific issue. Another way to solve this is to ensure there are no space characters in any paths to input, output or map files.

Errors

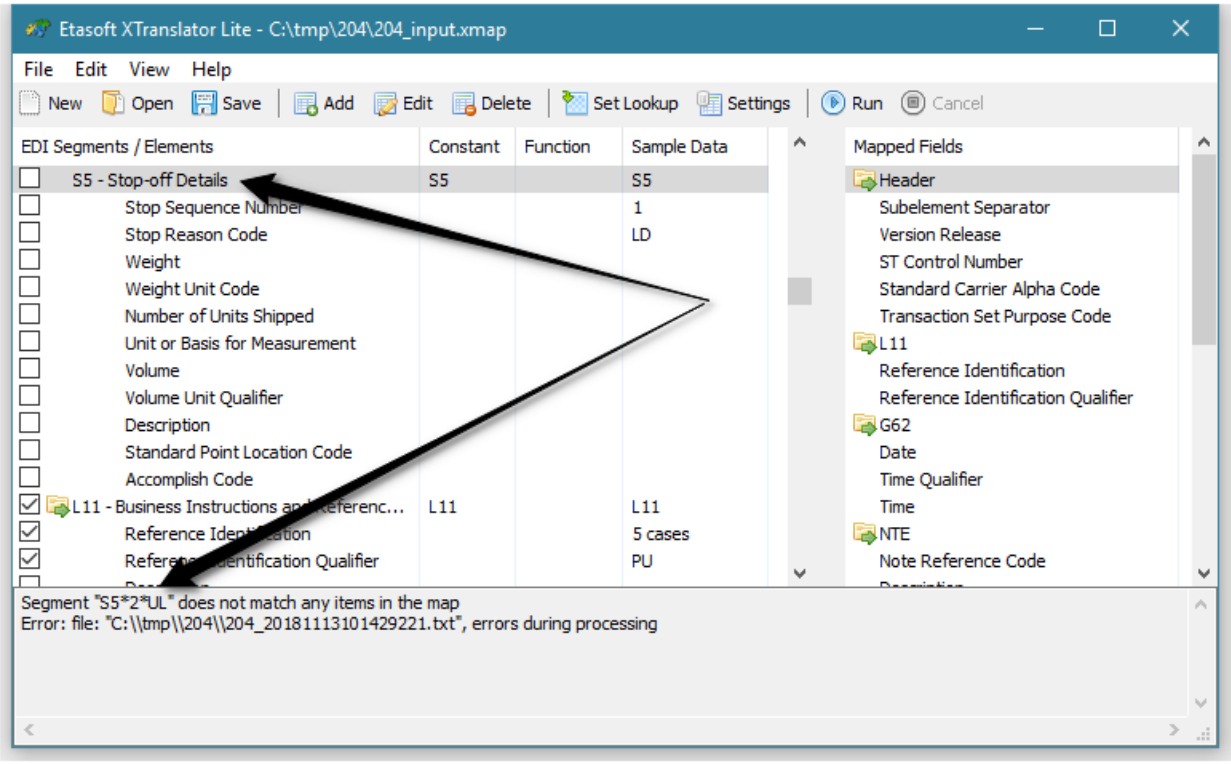
Translator displays mapping errors in the Map Editor's bottom panel. Typical errors are:

1. EDI segment is in the input file but not in the map.
2. Segment is in the map but it is setup in the wrong loop.
3. Segment is setup in the map but qualifier set on one of its elements does not match input. Since qualifier does not match, whole segment is not processed.
4. If input element is mapped to the loop that does not repeat as many times as that element. Translator does not have a place to put extra values because field already has a value.

It is important to build map step-by-step. Your workflow should be: map few items, test run the map, map few more items, test run the map, add segment, test run the map, set qualifier, test run the map, etc.

Every time you encounter the issue while running the map undo your last change. It also helps to save different versions of the working map during the mapping process. If problem occurs you can always go back to last working version of the map.

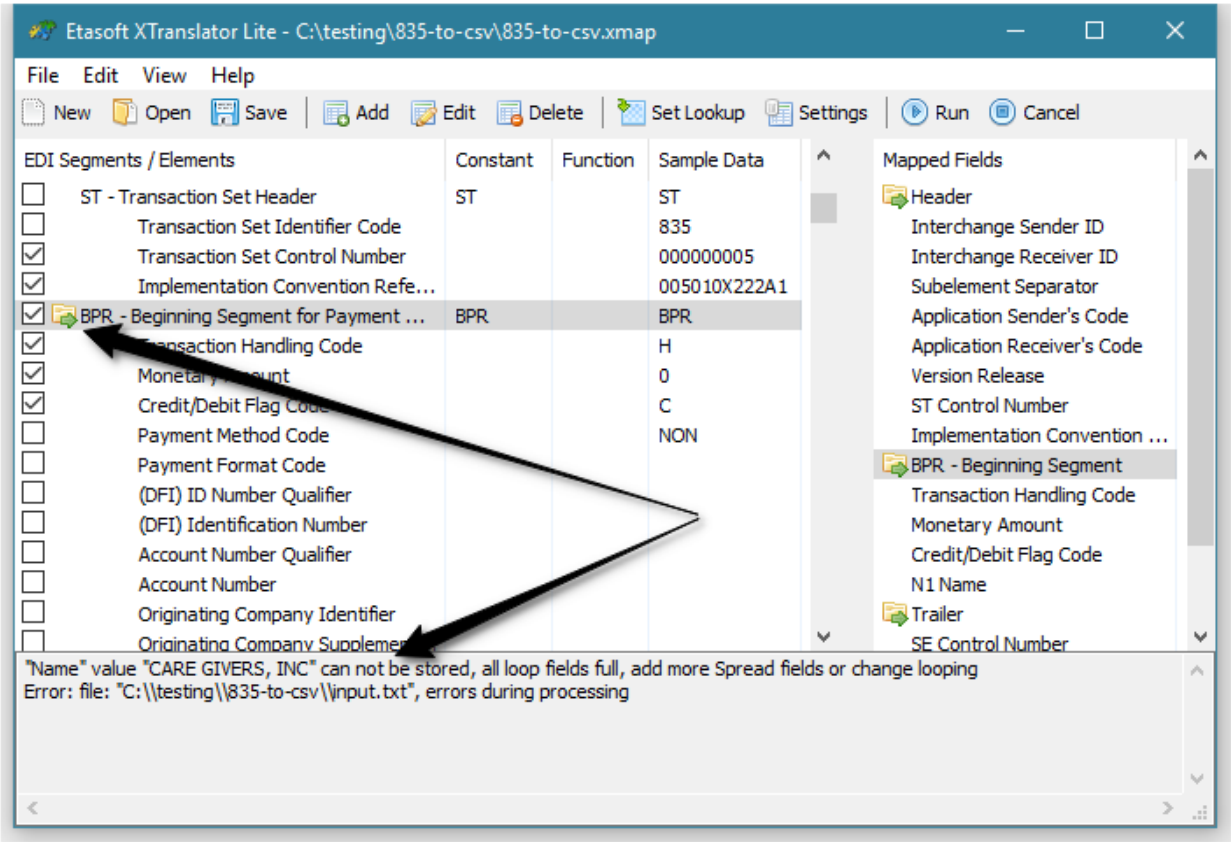
It is easier to start mapping top-to-bottom. So start with segments ISA, GS, ST and continue mapping down by adding extra segments with specific qualifiers.



In this contrived example "S5" segment is in the wrong loop. It should be setup below L11. Translator is unable to process it and displays an error.

Important point to remember when fixing translation issues processing input EDI file:

1. Translator matches segments by values in **Constant** column. Any qualifiers and segment ID listed in **Constant** column for the segment will have to match for segment to be processed.
2. But that is not all. Segment also has to be in the right loop. If it is in the wrong loop then it will not match and get displayed as error. This makes setting up loops an important part of the mapping.



Another common error is when there is not enough loops setup in the map. You have an element that repeats more times than loop that element is in. There is contrived example of EDI data:

Our input data	Our map
BPR segment N1 segment N1 segment	➔ BPR segment (Loop Start here) N1 segment

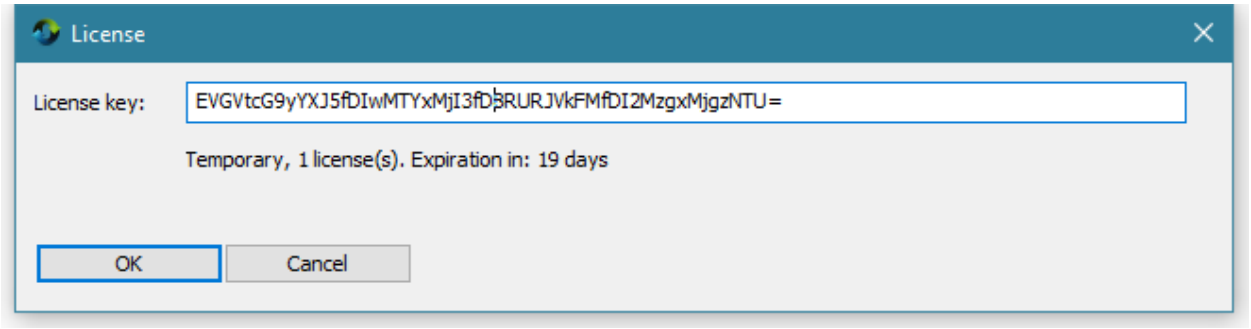
Our map only contains BPR and NM1 segments. Loop is set on BPR segment. But there is no loop set on N1 segment. In this case translator reads BPR, starts loop processing but then reads two N1s. Second N1 has fields that do not have a place because loop start is set on BPR and BPR repeats only once.

Solution: set loop start on N1 segment as well or add more N1 segments but with different qualifiers. Another solution is to use **Spread** and spread extra repeating values into fixed set of fields horizontally.

License

XTranslator Lite comes with trial and retail license. You can try and evaluate it free for 3 weeks. Once trail version expires please purchase license for non-trial use.

To start a trial version and receive license key for 3 weeks simply run GUI tool and open Help-License screen.



Use license key from GUI Help-License screen to run command line tool, and pass license into it via –license parameter.