



XTranslator guide on database to EDI X12 translation

Copyright © 2008-2018 Etasoft Inc.

Main website <http://www.etasoft.com>

XTranslator website <http://www.xtranslator.com>

Basic requirements 2

Software requirements 2

How to define database queries and fields 2

How to define EDI X12 layout 10

Review 19

Basic requirements

This document describes the process of mapping EDI X12 file to database for translation. For the mapping you should have:

1. Database with fields to produce EDI X12 file. In our case we use EDI X12 837 Healthcare Claim (X12 Release 4010). Sometimes you might not know all the fields you need to produce valid EDI X12. If you have sample EDI X12 file you can find out all the fields using our ediReports product. It simply prints all data with field names in CSV-Excel format.
2. Documentation explaining EDI X12 message layout and structure. You should have some document that would list all the EDI X12 segments and elements that you need to create in the EDI X12 file. Contact your trading partner or supplier for sample documentation on EDI X12 message you need to translate from the database. This documentation usually lists some specific required segments and provides EDI X12 message number and release version number.

Once mapping is done you do not have to recreate it again simply save it into the file with extension *.xmp. You can run map files using other utilities that come in the package (read User's Manual about other utility programs).

This document comes with complete map in a file with extension xmp. Please open and examine it as you read this document.

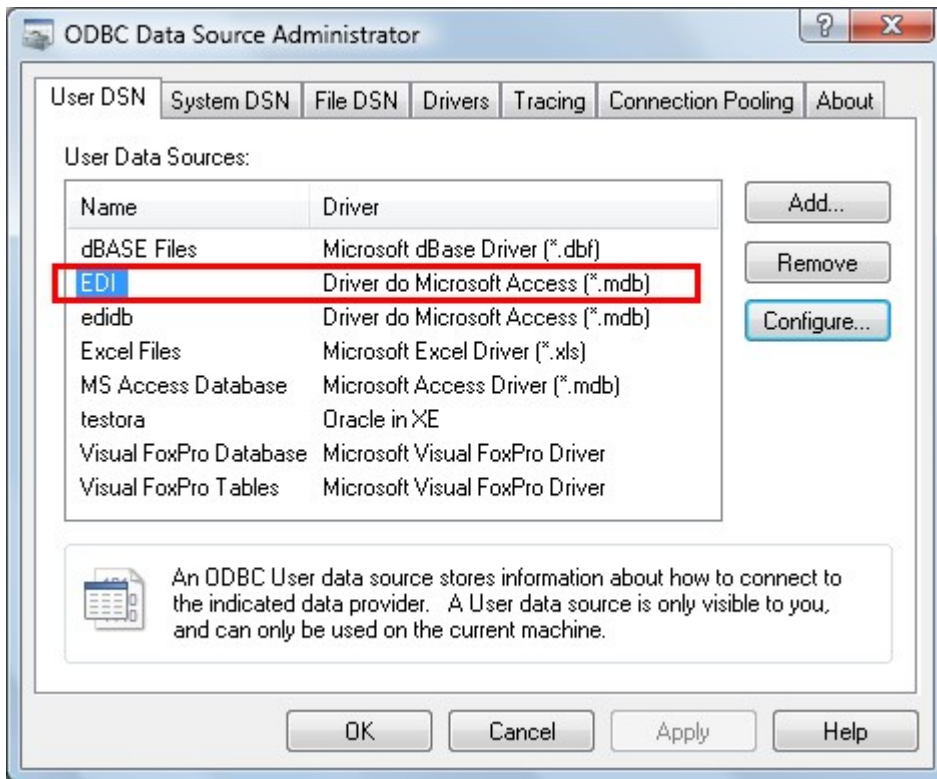
Software requirements

You will need to download and install XTranslator from the website <http://www.xtranslator.com> . Translator comes with number of templates accessible via Template Wizard. Setup program asks if you want to install templates that come with the package. If you choose not to install them, Template Wizard will not work and you will not be able to follow this document.

Once it is done, start Map Editor tool.

How to define database queries and fields

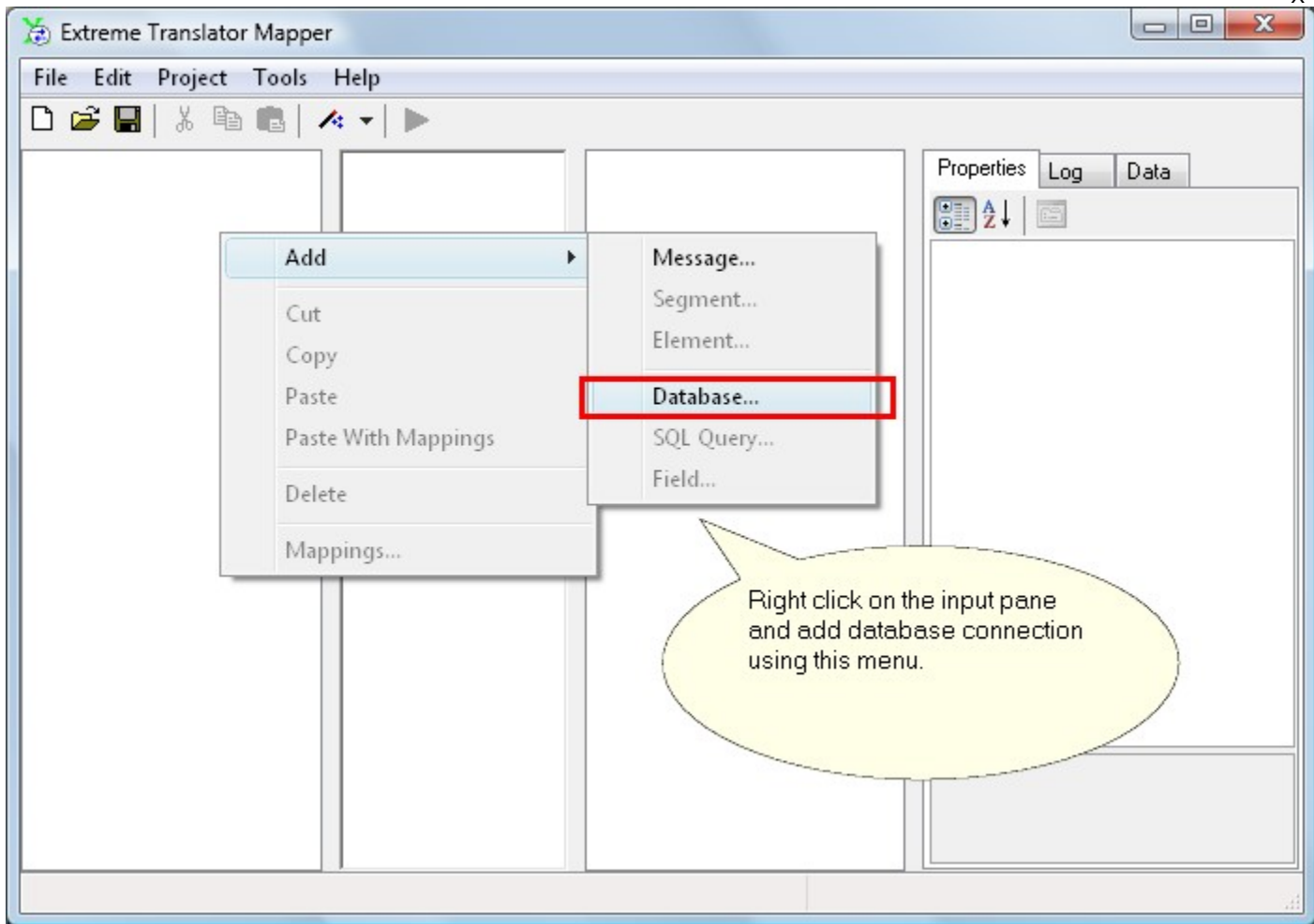
Before starting working in the Map Editor add your database connection to the list of ODBC connections in the Control Panel.



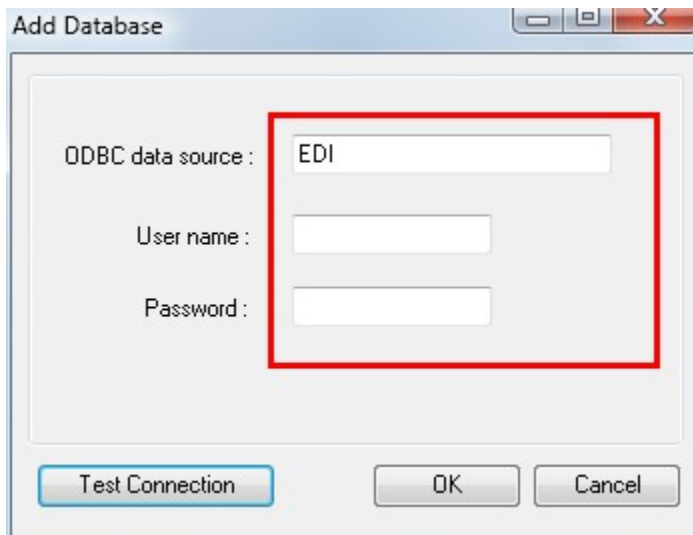
Add ODBC connection using Add button in Control Panel applet.

If you have Windows 64bit OS then make sure ODBC driver is also 64bit driver. Translator cannot work with 32bit ODBC drivers on Windows OS 64bit. Basic rule: ODBC driver bits have to match Operating System (OS) bits.

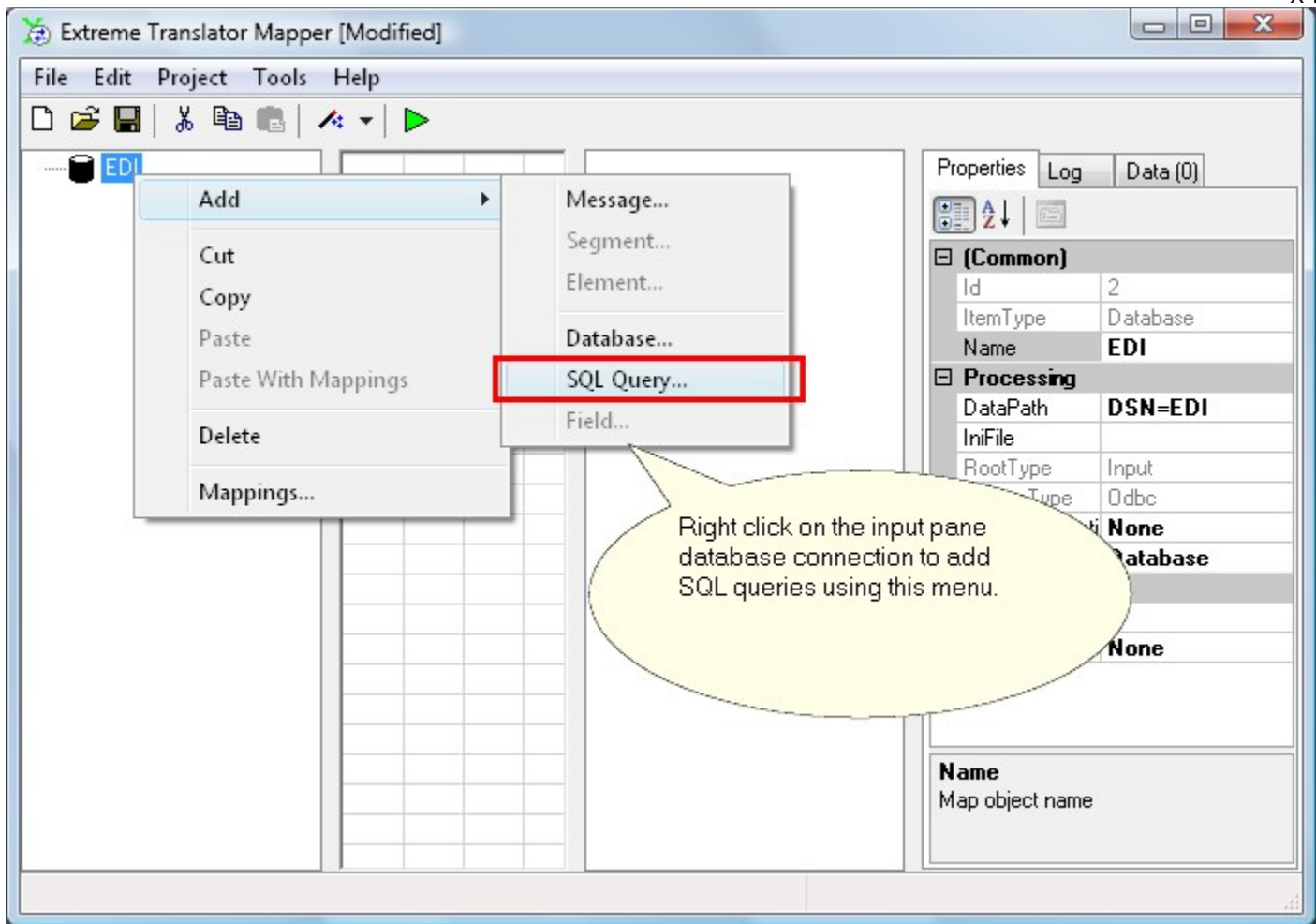
If you try to use 32bit ODBC driver you might be able to add it to Control Panel but driver may not work from the application on 64bit Windows OS.



Add Database connection via Map Editor.



You must define ODBC connection in Control Panel prior to using Test Connection button.



Add database SQL queries.

A frequently asked question is: how many queries or tables do I need on the input side to generate EDI X12 file? Each implementation is unique and so each translation map is unique. If you already have database structure defined your options are limited.

We recommend approach that usually gives good performance and is easy to understand: create as many SQL queries as many main loops you will have in EDI X12 file.

Some of our customers go to extreme by creating separate query for each EDI X12 segment they will output. This will work but will result in many separate SQL queries and usually more queries you have slower database connection gets.

In other words, more separate queries – more work for the database to fetch all the data. Granted you will not feel performance difference on few small EDI X12 files. But if your goal is to produce lots of big files performance turning is important.

We recommend having as few queries as possible.

In this map we only use 3 queries: header, detail and detail2. NM1 segments do not form separate queries but are mapped to different fields in the each query depending on the main loop NM1 is in.

SQL Select Query

Options

Name : EDI837PHeader

SQL

SQL select statement :

select * from EDI837PHeader

☒ Interchange_id
☒ Interchange_Control_Header_Authorization_Inform
☒ Interchange_Control_Header_Authorization_Inform
☒ Interchange_Control_Header_Security_Informatio
☒ Interchange_Control_Header_Security_Informatio
☒ Interchange_Control_Header_Interchange_ID_Qu
☒ Interchange_Control_Header_Interchange_Sende
☒ Interchange_Control_Header_Interchange_ID_Qu

Read Fields

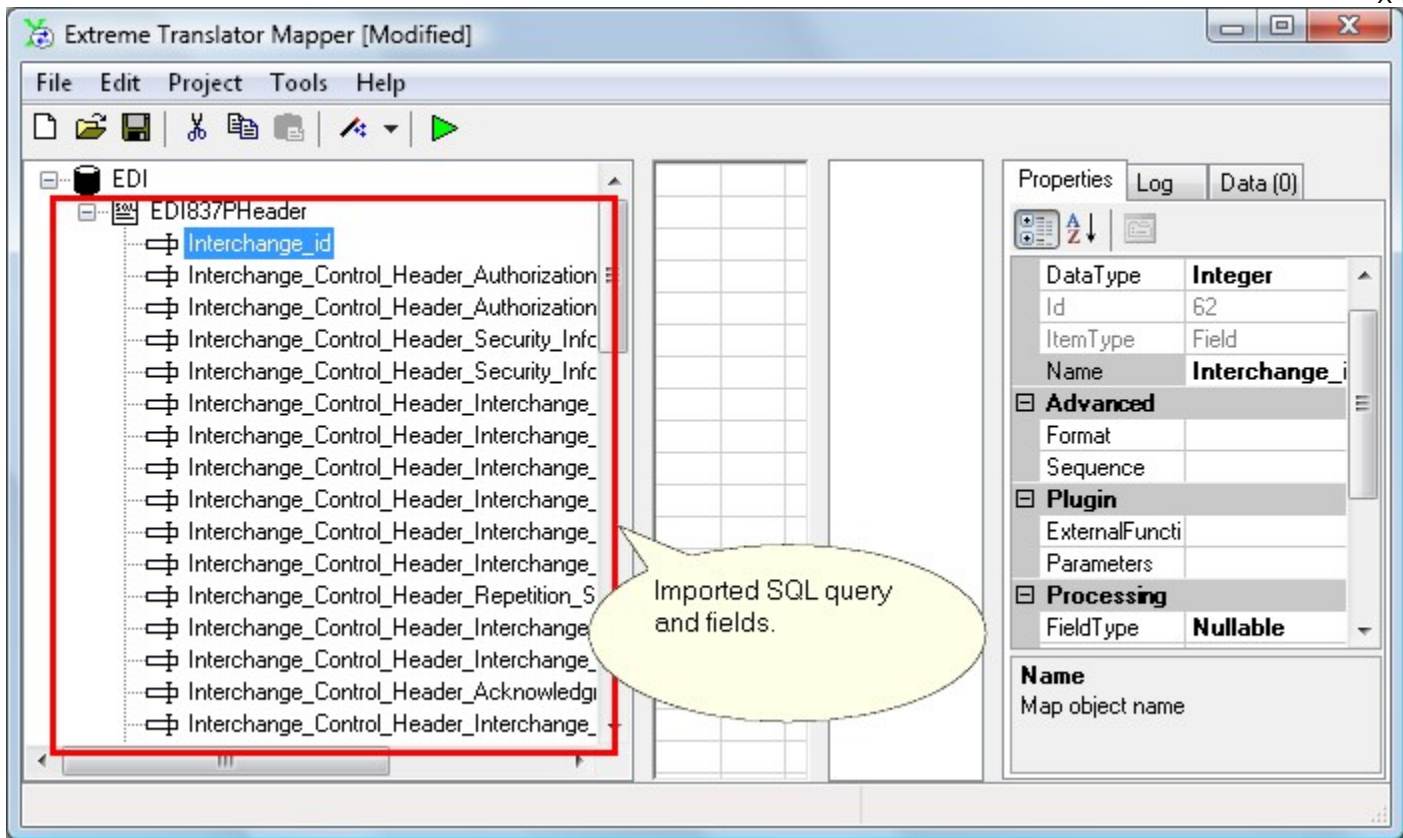
Check All

Uncheck All

OK Cancel

Import database query fields into the map.

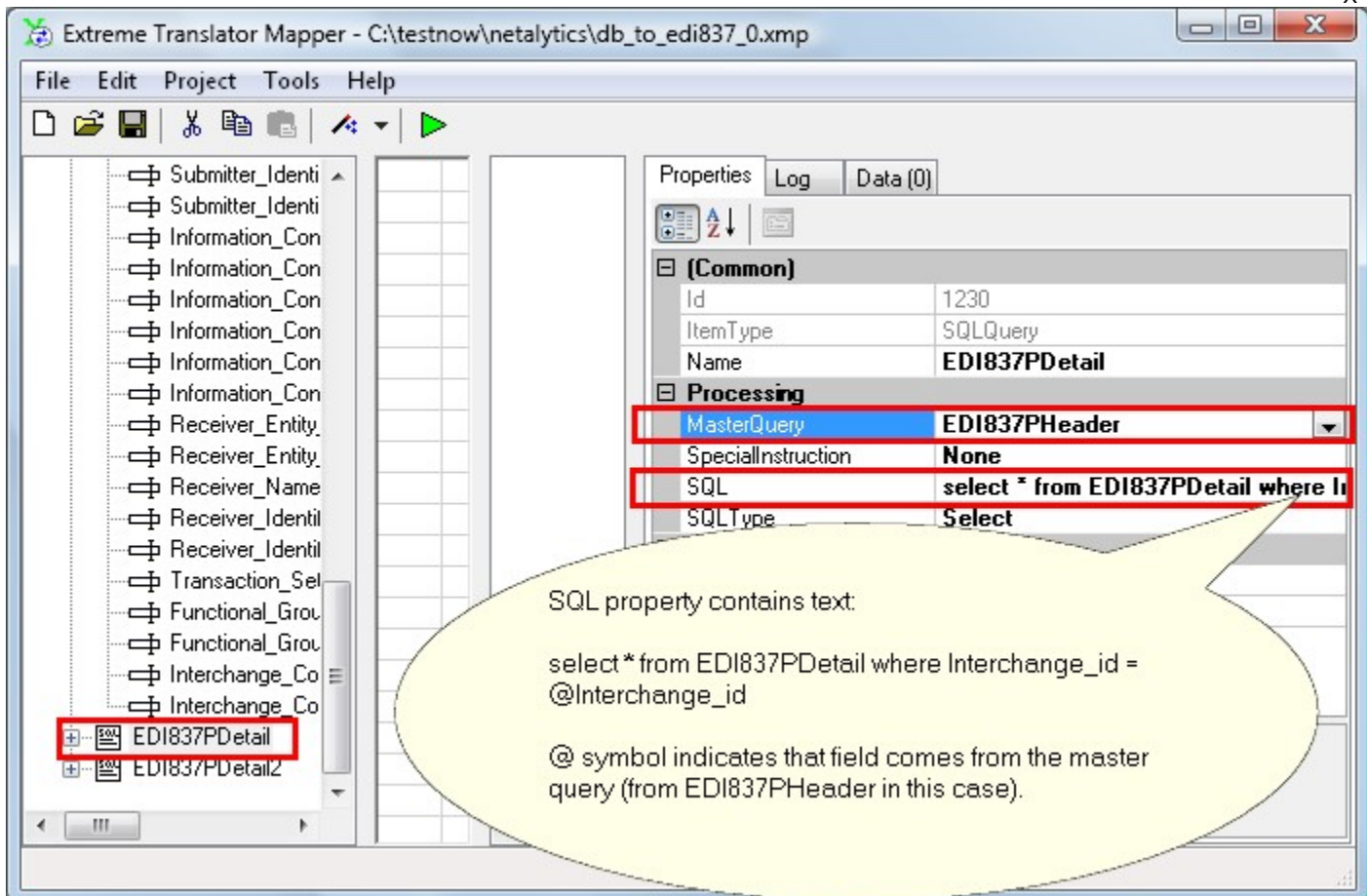
You can import table fields for each query. This wizard will attempt to read table structure from the database schema, and set field Size, DataType and FieldType properties for each imported field.



Imported SQL query and fields.

Once you import all queries and they fields you can setup master-detail relationships between them. Most EDI X12 files have looping structure. You will need at least few queries or more to produce this structure.

Translator supports master-detail queries, where one query is executed to fetch header data then sub-queries are executed repeatedly to fetch details and sub-details in order to form EDI X12 loops. Header query is called master and related queries are detail.



In this example EDI837PDetail is first detail query. It is setup with MasterQuery property pointing to the master and with link inside SQL property. Link is established with symbol "@".

Example detail SQL:

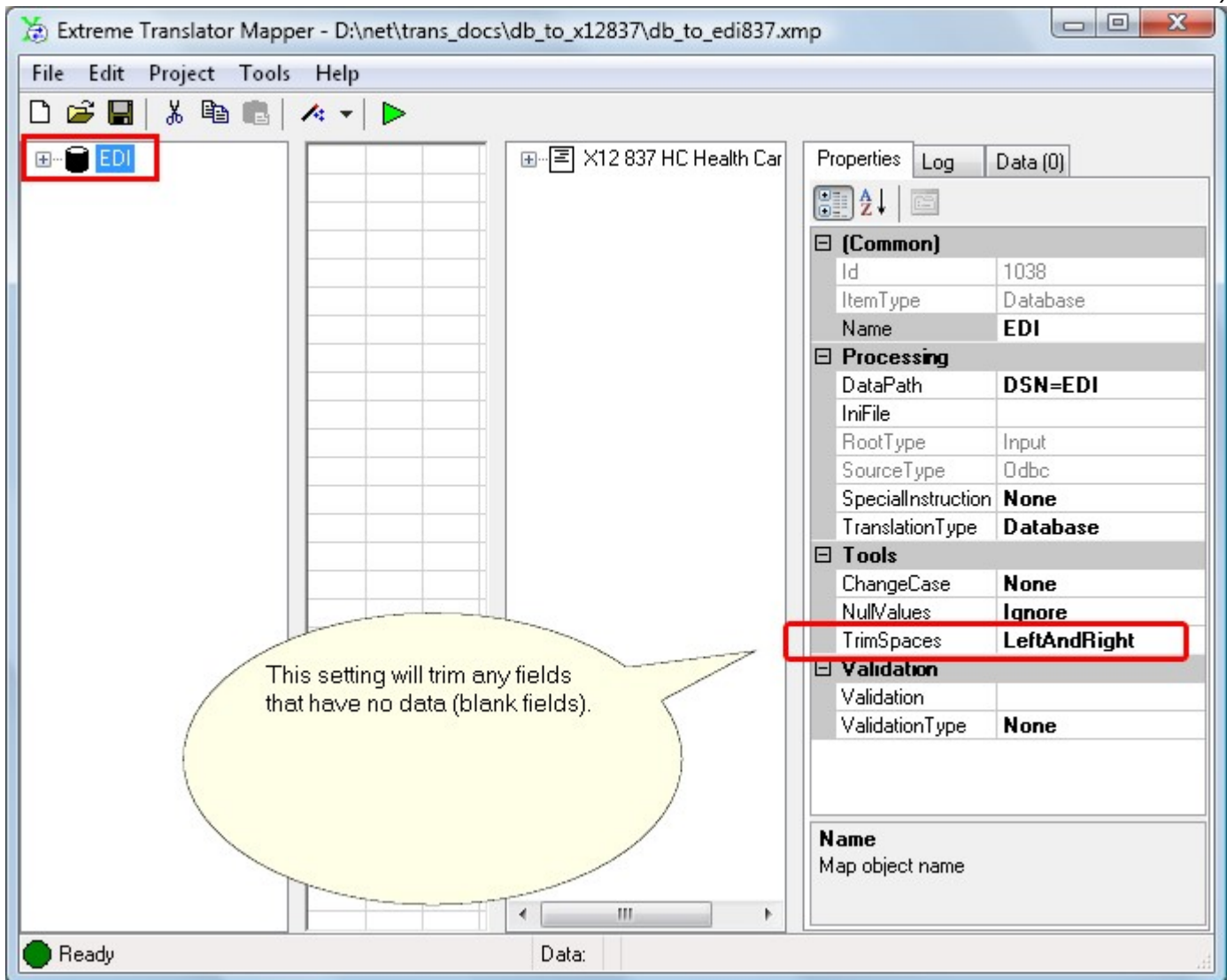
```
select * from EDI837PDetail where master_id = @master_id
```

Field master_id must exist in the master query. It does not have to be called master_id, we use it as an example.

Usually master-detail links are logical and based on primary key and foreign key settings between database tables. You can also have master-detail-detail2 setups where master links to detail and detail links to sub-detail.

You can have queries nested many levels deep. But keep an eye on possible performance issues. Each nesting level will cost you processing time. More levels you have – slower it gets. This is because translator has to cache records in order to perform repeated execution of the nested queries.

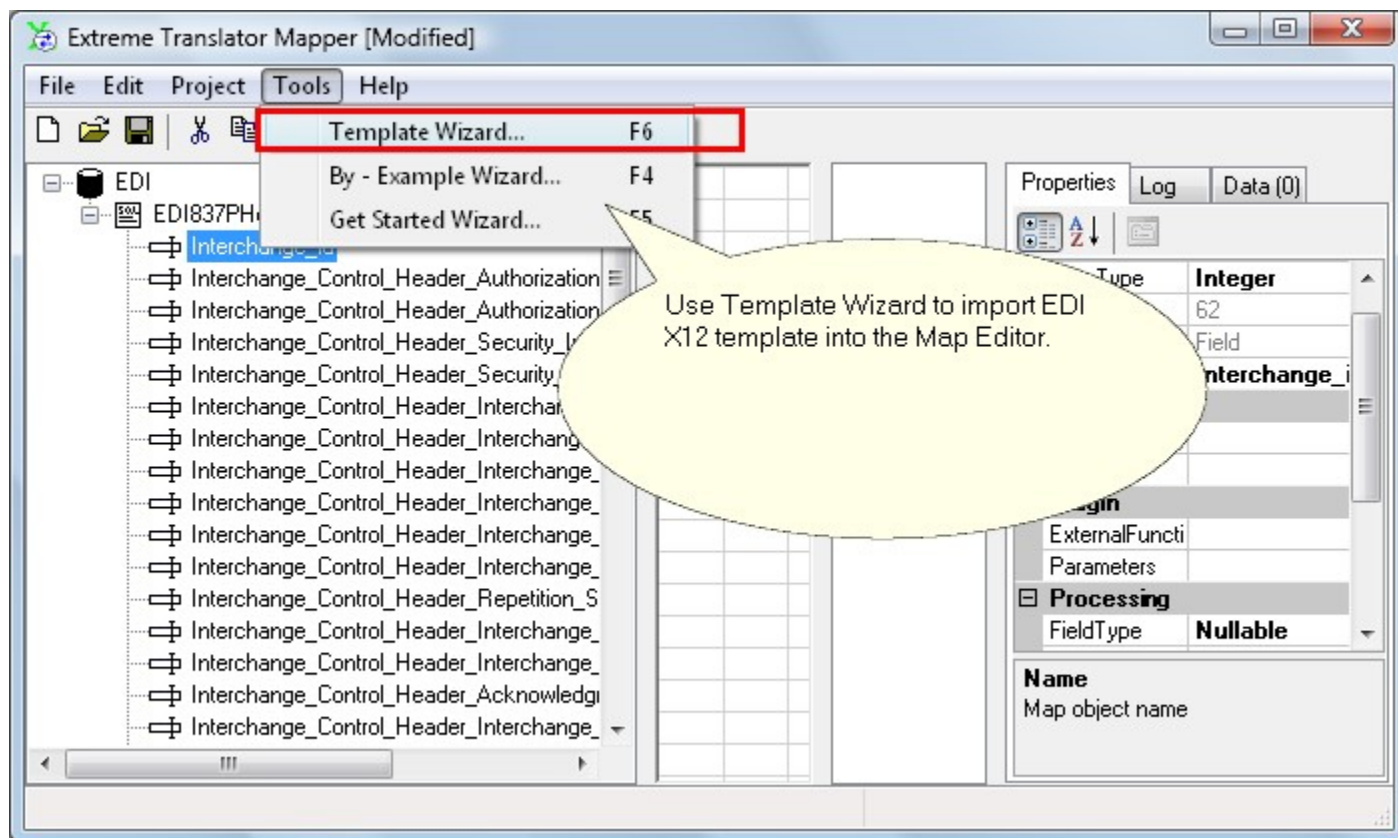
Almost all EDI X12 message types can be produced by having only three queries: master, detail and sub-detail. EDI X12 837 is no exception. Three queries should be enough to fetch all the data for EDI X12 837.



In some cases you may want to trim excess spaces from fields and do not produce fields with no values in them. TrimSpaces setting does just that.

How to define EDI X12 layout

In the Map Editor under Tools -> Template Wizard menu you will find many templates for most widely used standard formats like EDI X12 (X12N), EDIFACT, SAP and XML. Most templates come with setup package, just make sure to install them once setup program gives you a choice in a form of checkbox. If you will not install templates, then \templates subfolder will be missing under translator installation folder and Template Wizard will not work.



Open Template Wizard using this menu option. Select specific template and press Import button.

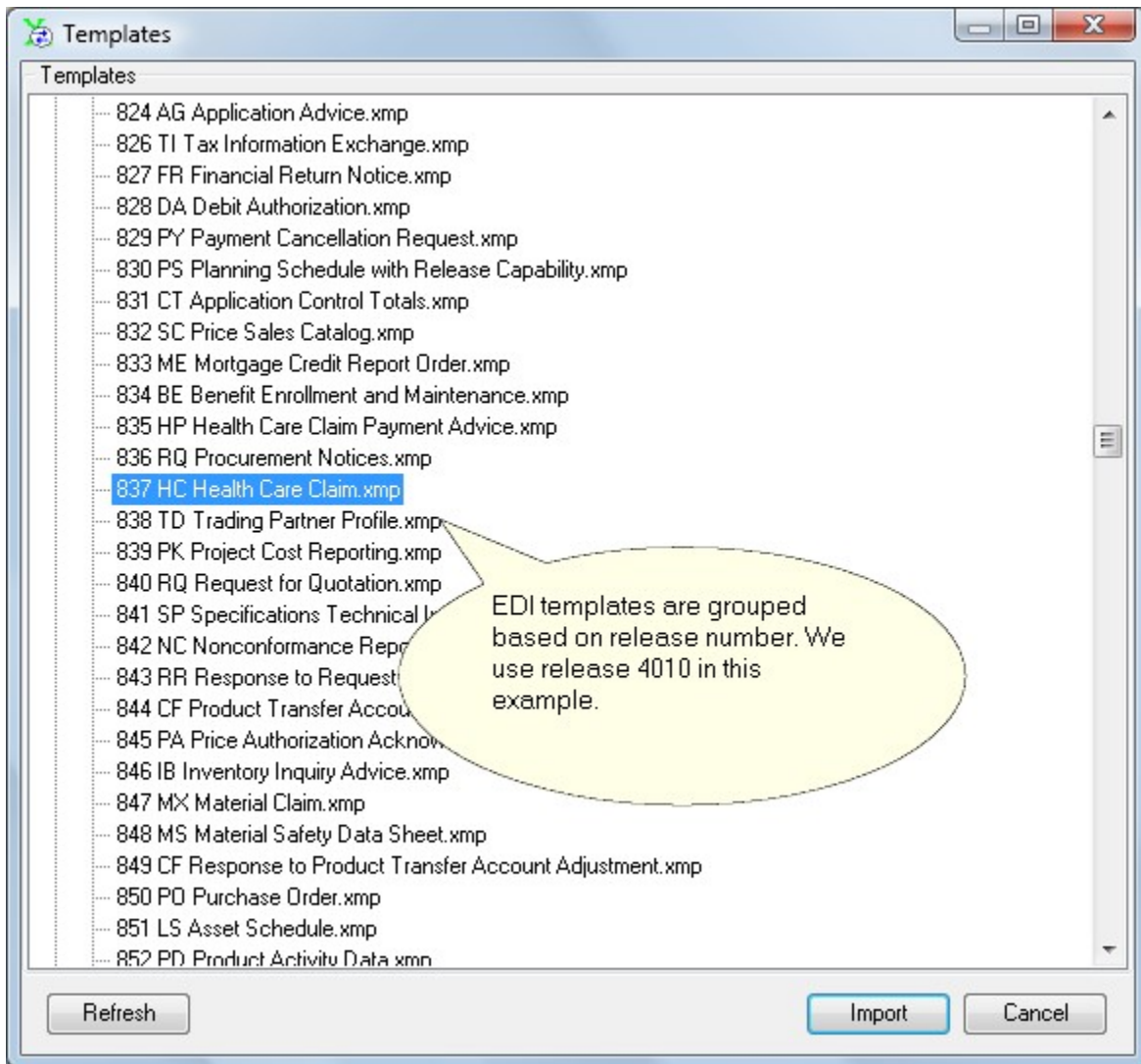
You can import EDI X12 layout using template from Template Wizard. Many EDI X12 templates are included with the product. We update and add more templates with every new release of the product. All templates are stored in \templates directory. Templates are grouped by release version.

Translator includes templates for major standard releases. It may not have all the minor releases.

It is important to understand the difference between major and minor standard releases. In EDI X12 major releases numbered with schema "xx10" having last two numbers as "10" always. Translator has major releases 4010, 5010, 6010, etc. as templates but has no minor releases such as 5040 or 6030.

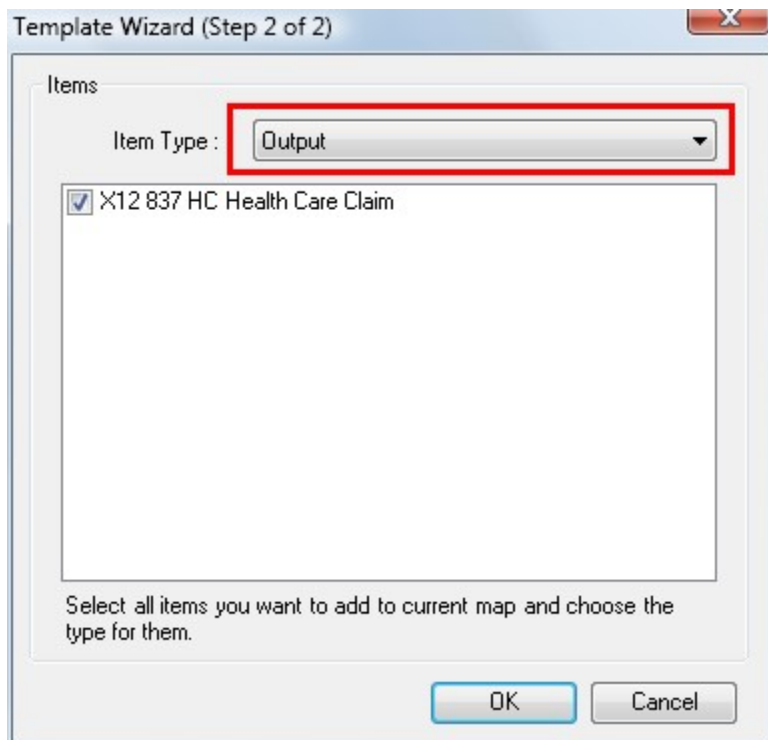
Most of a time only major releases are used for EDI implementations. If your trading partner requires minor release you can simply use major release as substitute. Example: use 5010 instead of 5040.

Some very small changes to the template might be required in order to adjust template for minor release.



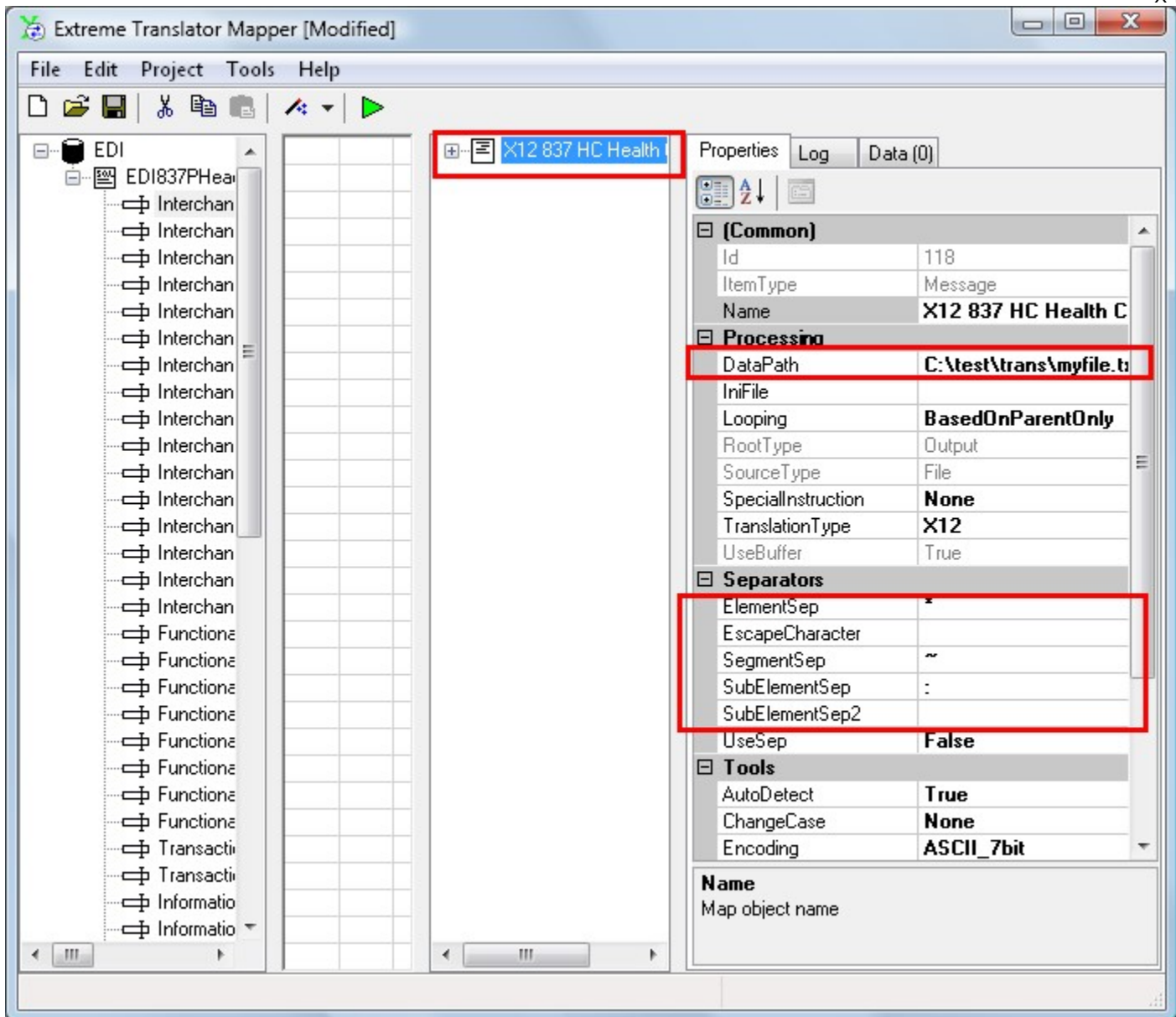
Import EDI X12 message. In our case we import EDI X12 837 release 4010.

You can import templates for input and output side. Second dialog screen in Template Wizard will ask if it is input or output template import.



Choose template from the list.

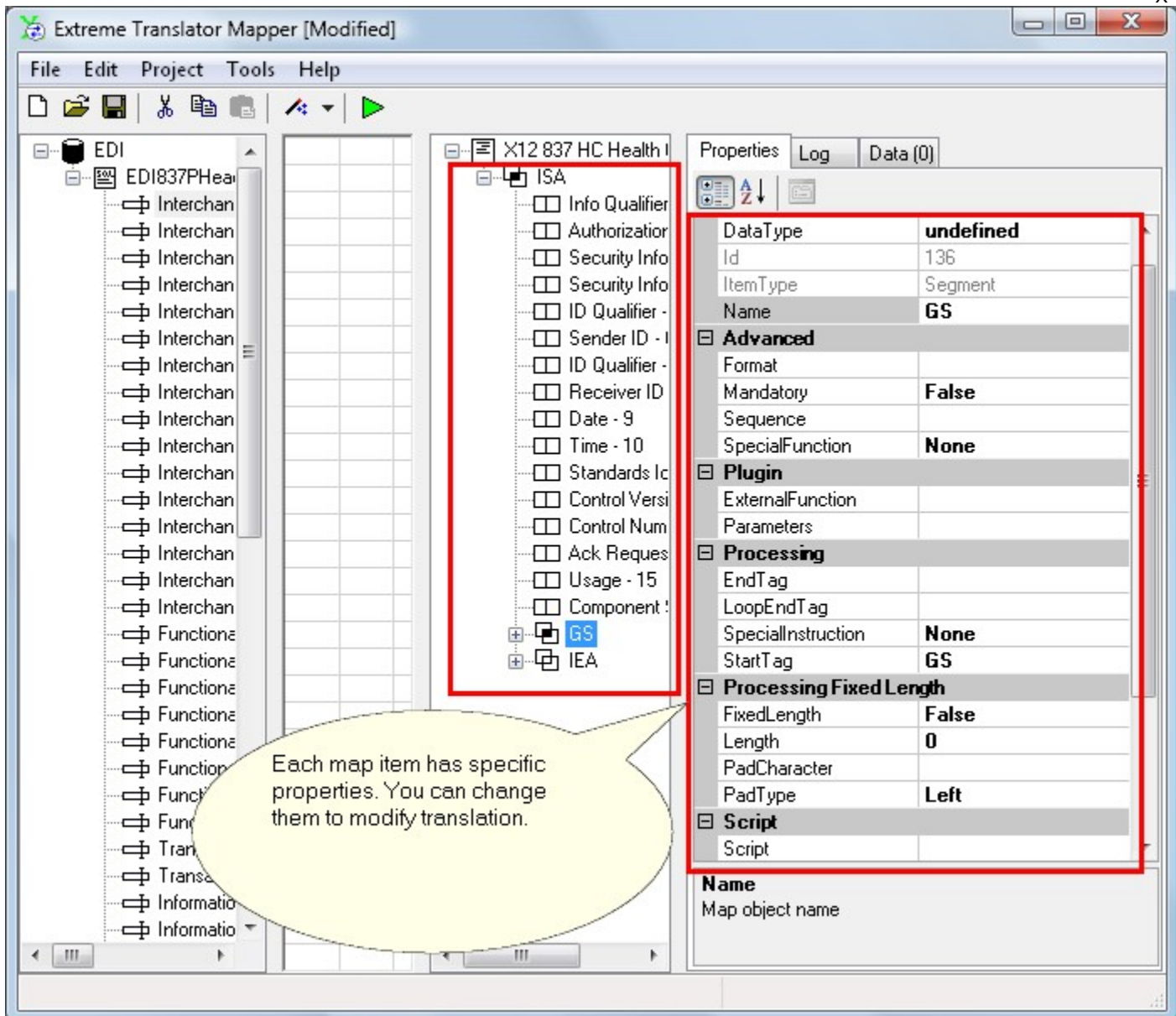
Once you import template, it will load number of map objects. You will see object called "message" or "file" at the very top of the mapping tree. This object holds properties such as input or output file name, separators, number of other useful tools.



Root item in the map is called “message” or “file” and it contains most important properties related to file processing like input file name, segment separators, etc.

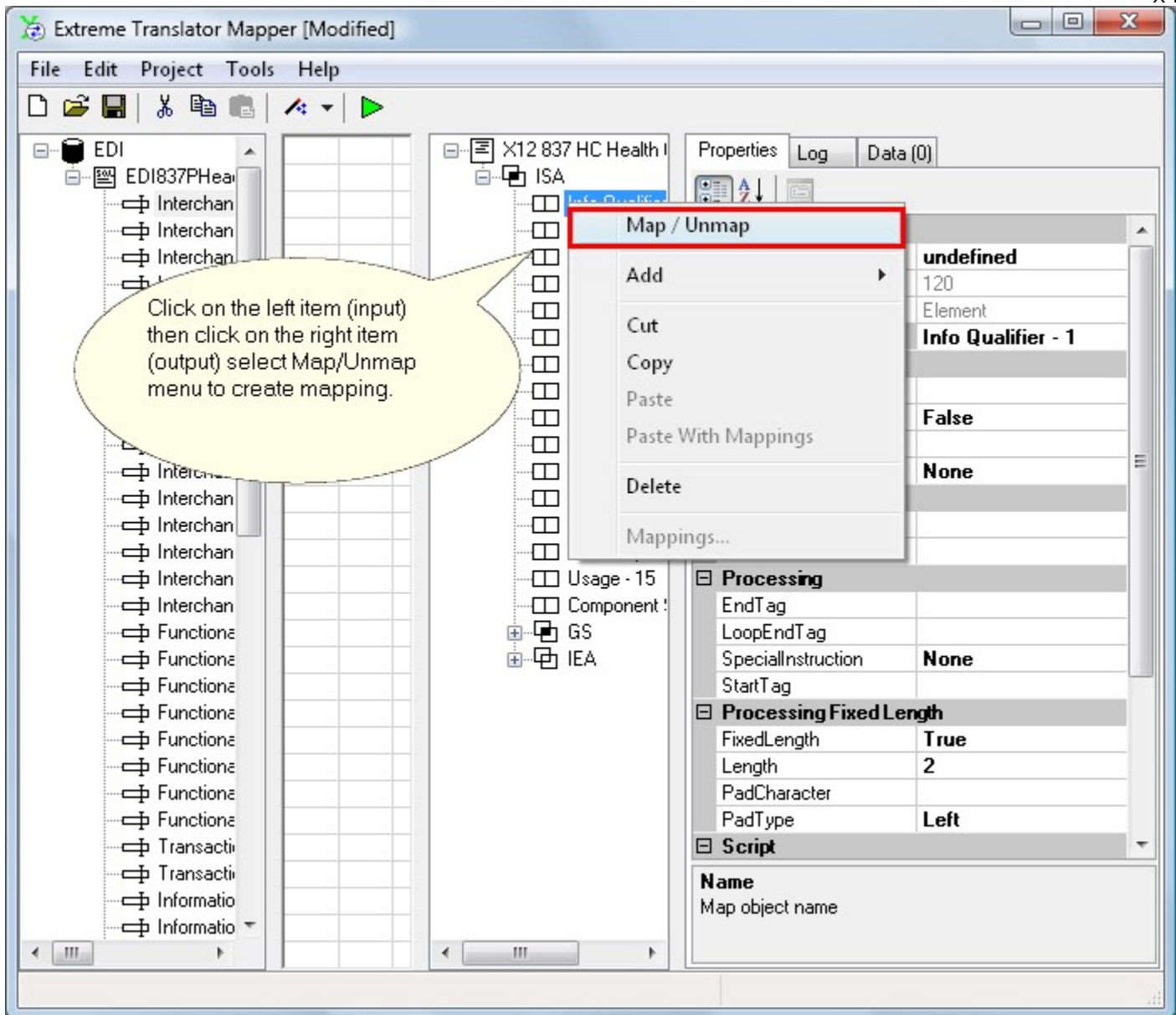
EDI X12 files may contain all kinds of separators from most typical “*” (star), “~” (tilde) and “:” (colon) to less typical non-printable characters. Check with your trading partner to find out what separators they expect in your file.

DataPath property points to your output file that translator will produce once translation runs. This value can be changed at runtime. You can also use macros to produce dynamic file names with dates, times and counts embedded in the file name. Check Translator Users’ Manual for DataPath property.

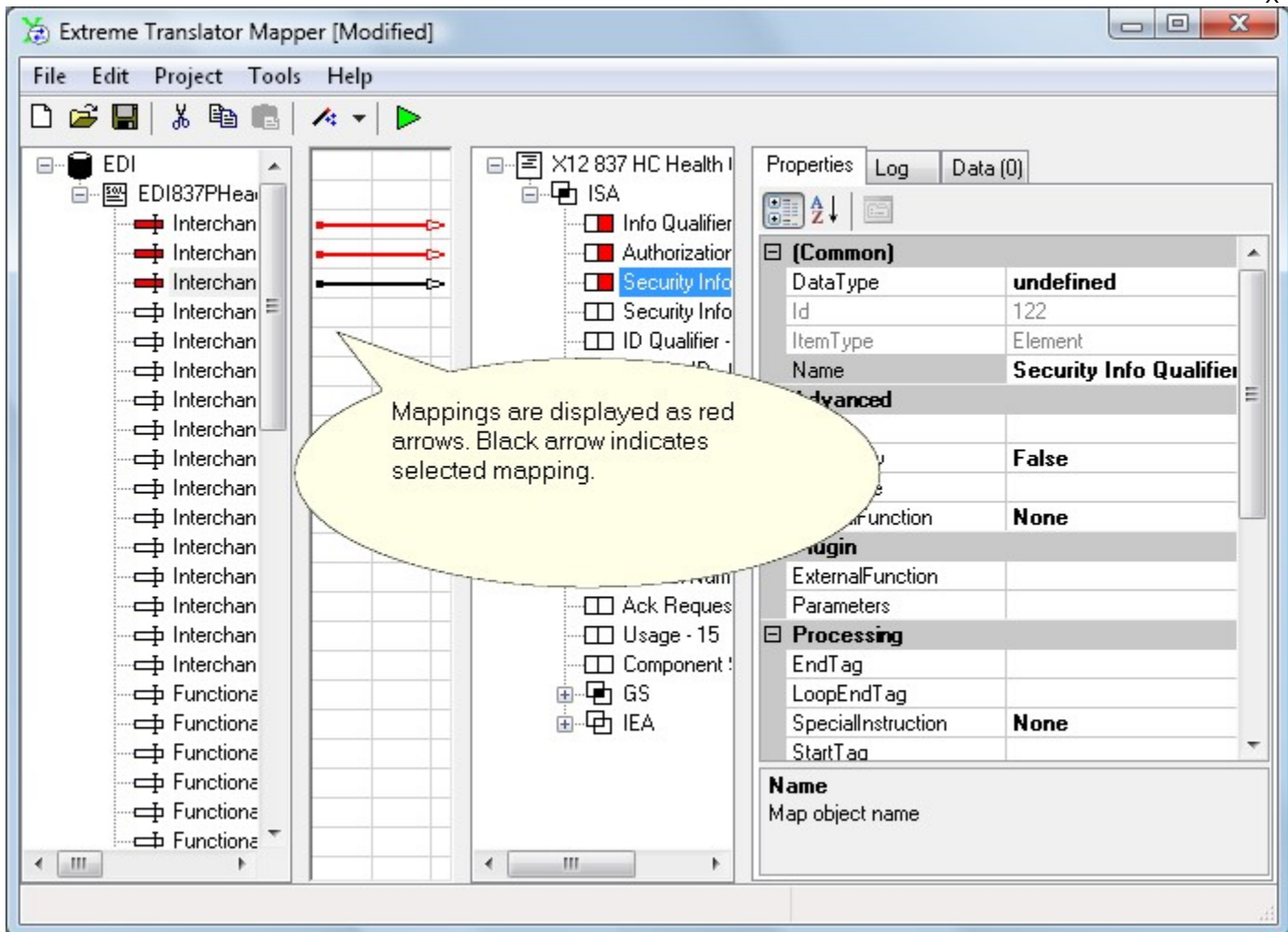


You can click on any item in the output side and see the properties for it listed on the right side of the Map Editor window.

You can save and run unfinished map. It will only read incoming data but will not output anything because mappings to the output side are not defined yet.

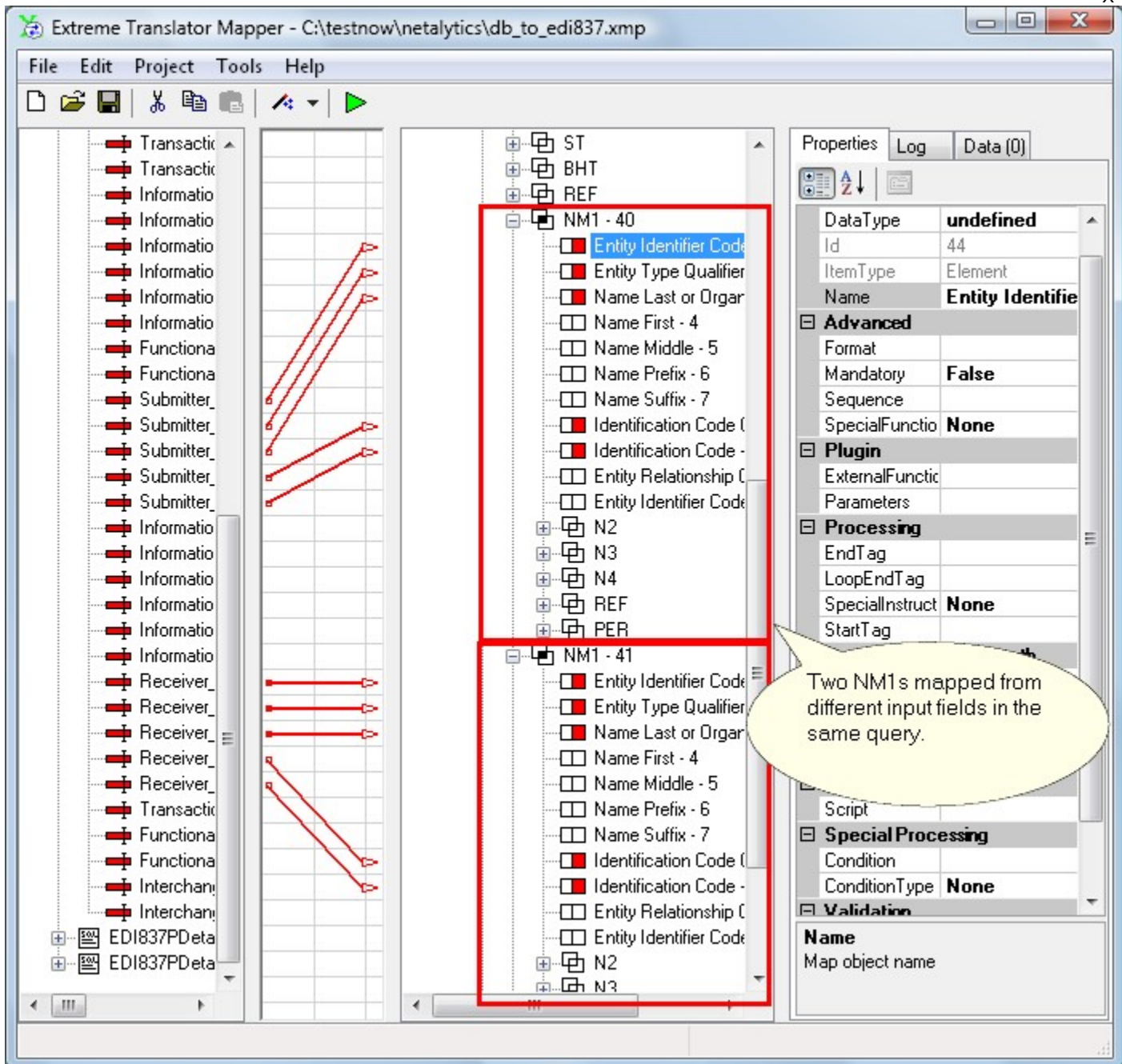


Create mappings using Map / Unmap menu.



Once you select items that have mapping between them, that mapping changes color to black indicated that it is selected.

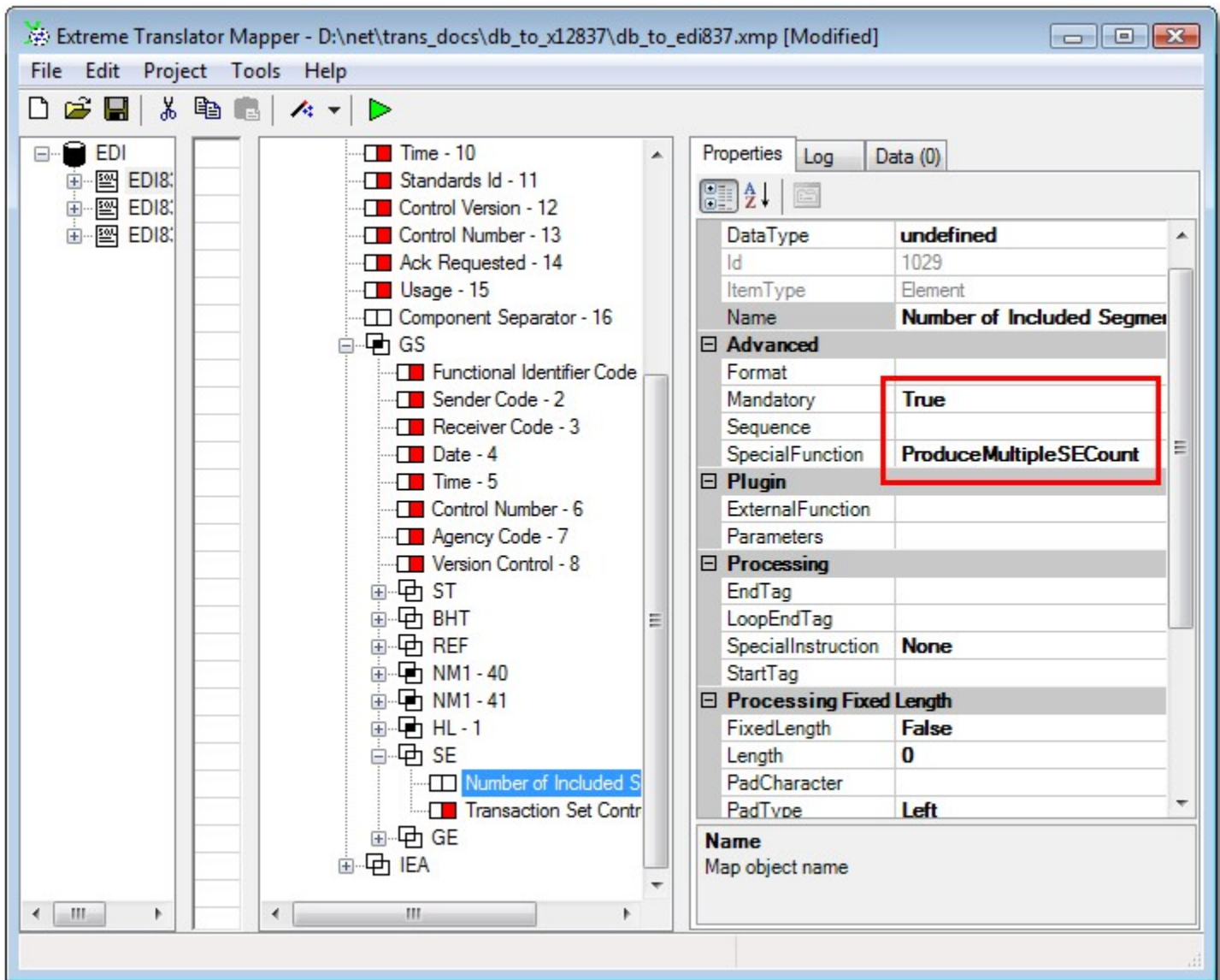
As you map more and more items make sure to run map and test results as you go along. This helps to ensure that work is progressing in the right direction.



In this map we did not separate NM1s into separate stand-alone queries. Instead NM1s are mapped to the same query of the main loop.

NM1 segments are mapped to the same query even if they repeat. Since each NM1 has different qualifier value each is unique for duration of the loop. We simply selected NM1 and used Copy & Paste menu to create one NM1 after the other. That way each can be mapped to different set of input fields.

All our elements are mapped except segment count. That is first element in SE segment. This value depends on how many segments should be produced on the output EDI file. Since value is dynamic it has to be calculated by translator. It is done by setting special function on SE element 1 and setting that element to Mandatory = True.



Example of setting segment count to generated value.

Run the map using green arrow button in the Map Editor toolbar.

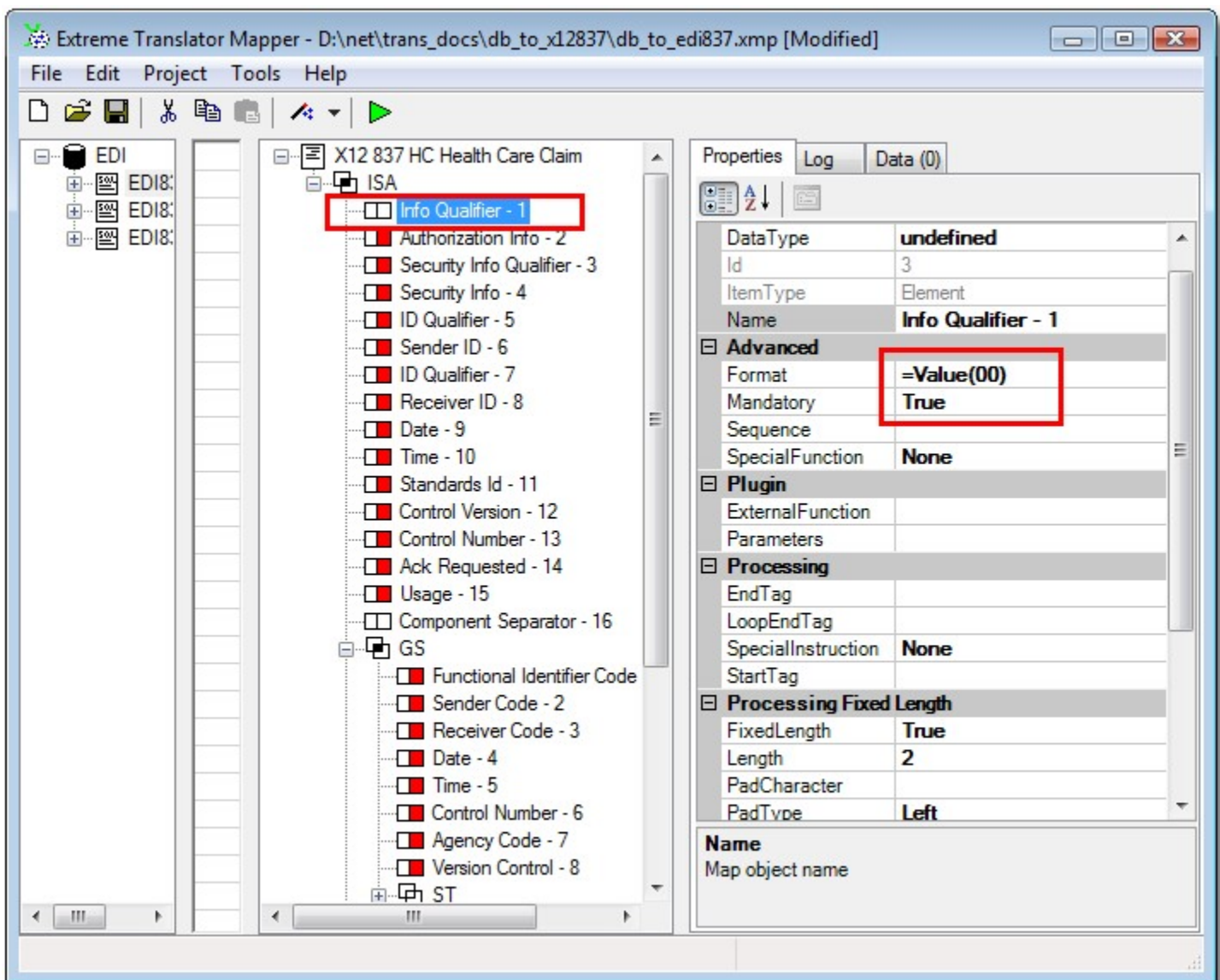
Review

Mapping process comes down to following basic steps:

1. Create ODBC data source in Control Panel.
2. Add connection to the map.
3. Add all SQL queries.
4. Create master-detail links between queries.
5. Import EDI X12 template.
6. Create mappings from input fields to output elements.

There are some advanced steps that we did not use in our map. We will cover them below.

In some translations you might need to place constant values in the resulting element instead of mapping it.



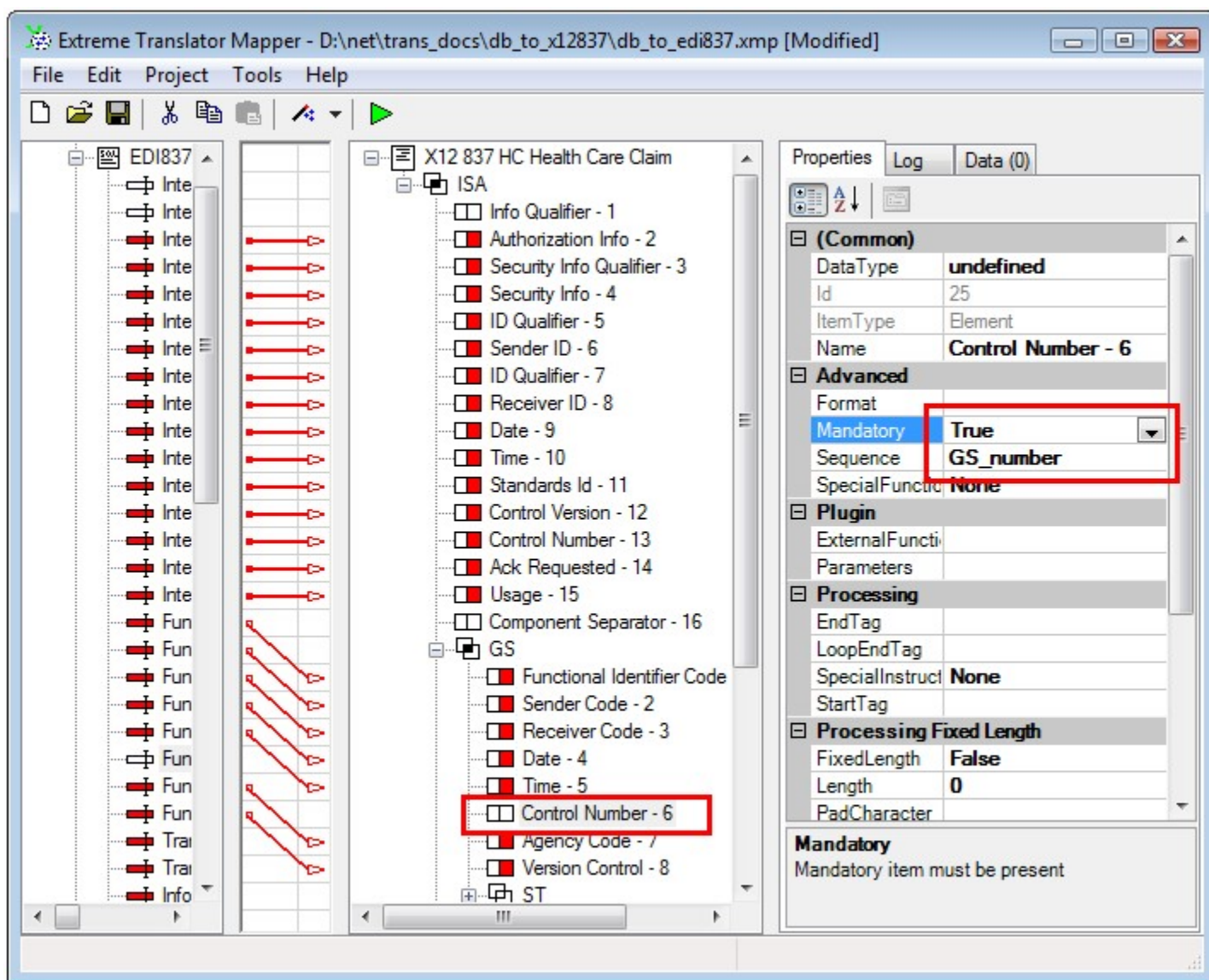
There is an example of using mandatory constant value instead of mapping it. Format property contains function =Value(_your_constant_data_) that is used to produce constant in the output.

If segment has no mappings to its nested elements and segments then it will not be produced in the output side. You can force such segment to be produced by setting it to Mandatory=True. If you also would like to output first few elements with fixed constant values make sure to set those elements to Mandatory=True as well.

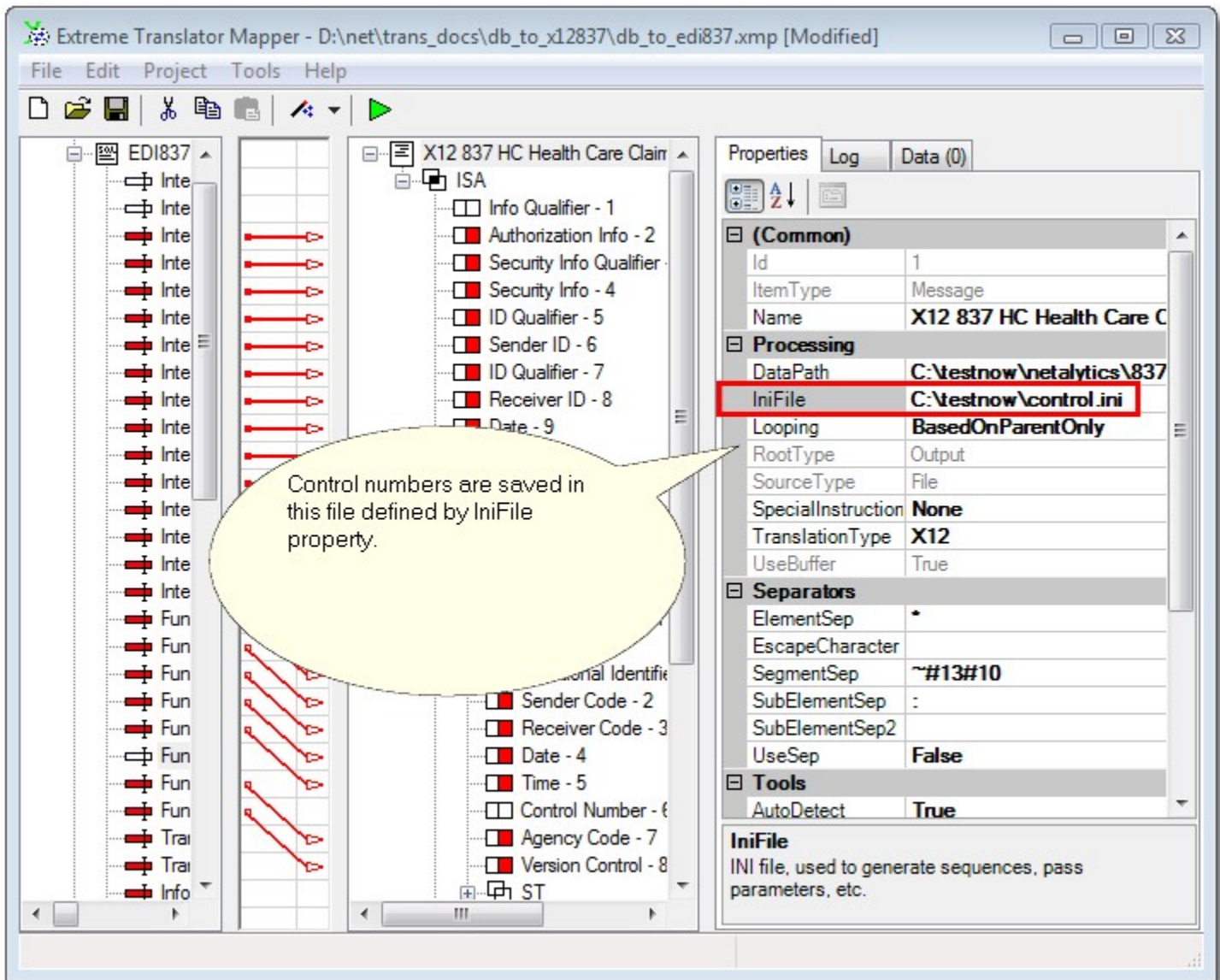
Basic idea is that Mandatory=True forces segment to be produced in the output even if segment has no mappings inside. But in order to also output elements of that segment you need to set them to Mandatory=True as well.

If you both map elements inside of the segment and also mark segment as Mandatory=True that segment will be produced twice.

Translator is able to generate control numbers for EDI X12 envelope segments. In this map we mapped all control numbers from the database fields. You can change the map to generate control numbers.



In this example control number generator is setup on GS element number 6 in property called Sequence. You can give number generator any name. In this case we named it "GS_number".



Because control numbers should always increment for every new file they have to be saved in the external file for storage. So next time translator runs it would use next available number. This is done by saving most recent value of control numbers inside the file set by IniFile property.