

Etasoft – Mini Translator 5.x

EDI Mini Translator is designed for automated and semi-automated translation of EDI X12 files into CSV (Excel). It supports all EDI X12 40xx, 50xx, 60xx, 70xx message types.

Setup

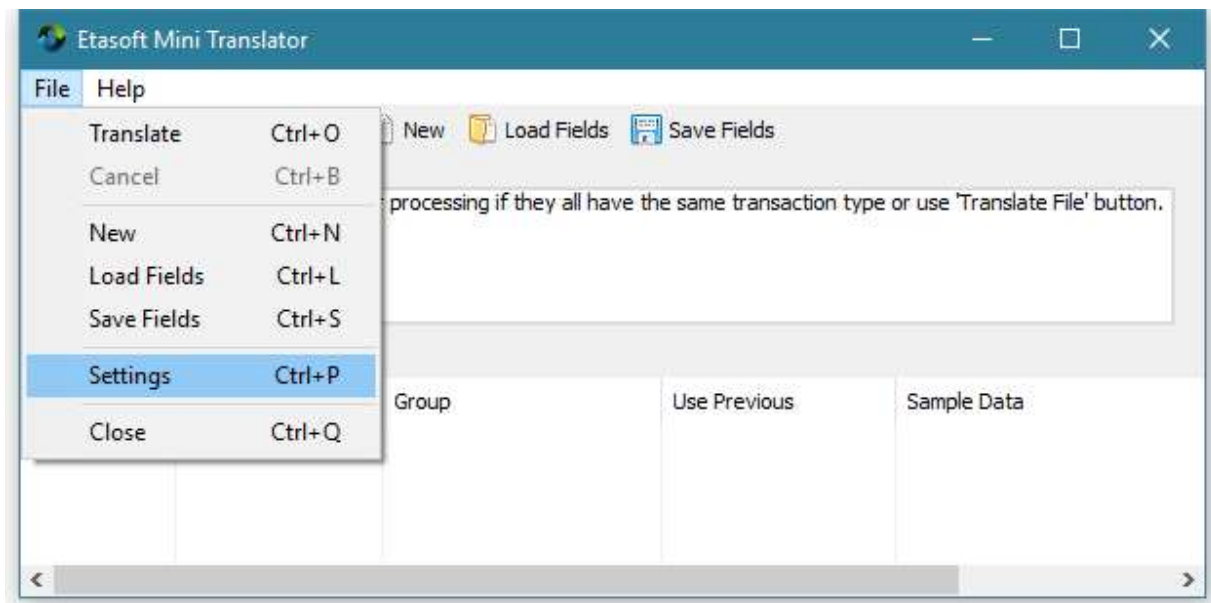
For Windows install the software using provided setup file minisetup.exe.

For Linux install the software unzipping the file minilinux.zip.

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

Processing

GUI based translation tool is designed for semi-automated processing. It can process single file or multiple files. Some of the additional options provided via Settings menu.



In GUI you can add or remove output fields and save configuration with fields into a file. Command line tool does not have these features but can load saved configuration file. So customize fields via GUI but run them via command line.

Translator produces CSV files ready for import into relational databases using custom import script. Sometimes it is useful to have data from EDI header loops repeated on each line. This helps line up header loop data with data from details loops. "Use Previous" setting does this. If turned on it copies data from prior row if new row does not have any new value for that field.

Avoid setting "Use Previous" on fields that might be optional. For example: date of birth field in some health claim formats. In some rare cases field gets omitted and data of birth of prior patient will get copied into the empty slot making it date of birth of next patient. Situations like that are possible. So use "Use Previous" only if you are sure that field is not optional.

You can change "Use Previous" setting by double clicking row in the list of fields.

Mini Translator aligns output rows based on EDI loops. It is designed to give you basic flat output to work with. Mini is minimalistic. There are no endless lists of options and features to tweak. Once data is extracted into flat format unleash full power of SQL or other scripting language to post-process and upload data into your systems.

Prior versions of Mini Translator included number of various approaches and options to line up fields in the output in certain order so they would be immediately importable into relational databases. This did not work as too many options made Mini hard to use and still did not solve all the problems. For full power you either need scripting language and some basic programming or good mapping tool.

So we removed the clutter and left only two options: use Mini and your post-processing scripts/tools or use full EDI translator that has mapping tools.

If you are looking for a tool to produce flat outputs that require no post-processing please take a look at our other products, such as XTranslator (Extreme Translator).

If you have many different trading partners you may have many EDI message formats that are very similar yet not the same. For example 837 Healthcare Claim format has number of variations that are not compatible. Same applies to many other popular EDI formats. You may have to use different sets of fields even when dealing with the same transaction-release numbers coming from different trading partners. Use 'Load Fields' and 'Save Fields' buttons to save fields specific to that trading partner.

Tips and Tricks

Mini Translator output might need to be post-processed because of the way EDI loops get arranged in the flat format. Once flat file is loaded into database use your favorite tools or programming languages to rearrange and clean up data.

In some cases EDI files have way too many loops to produce single decent flat CSV file. EDI files with HL

segments where loops are nested in deep structure like Russian dolls are hard to get flat. Sometimes it is easier to have few different sets of fields and run translator twice on the same input file to produce two CSV files. Then load each CSV into relational database and clean it up.

Always use “Save Fields”, “Load Fields” and run command line with `-fld` parameter if you want to get same set of fields. If fields are not loaded with “Load Fields” or `-fld` then translator performs “discovery” of fields when input file arrives. Discovered fields are added to CSV file and become part of the output.

So if you have few different trading partners and they use slightly different sets of EDI fields you will be getting different CSV for each one. Sometimes it is desirable. But if you want to have fixed set of fields, save and load them.

Sometimes EDI files have many differences when they come from different trading partners. This is especially true when specific EDI message is complex and many interpretations of fields exist in the industry. Again, use “Save Fields” and “Load Fields” to have fixed set of fields for each specific variation.

Create directory structure to help you process variations. Place each variation files into specific folder to help you process them independently. Good directory structure can go a long way in helping streamline processing.

Command Line Execution

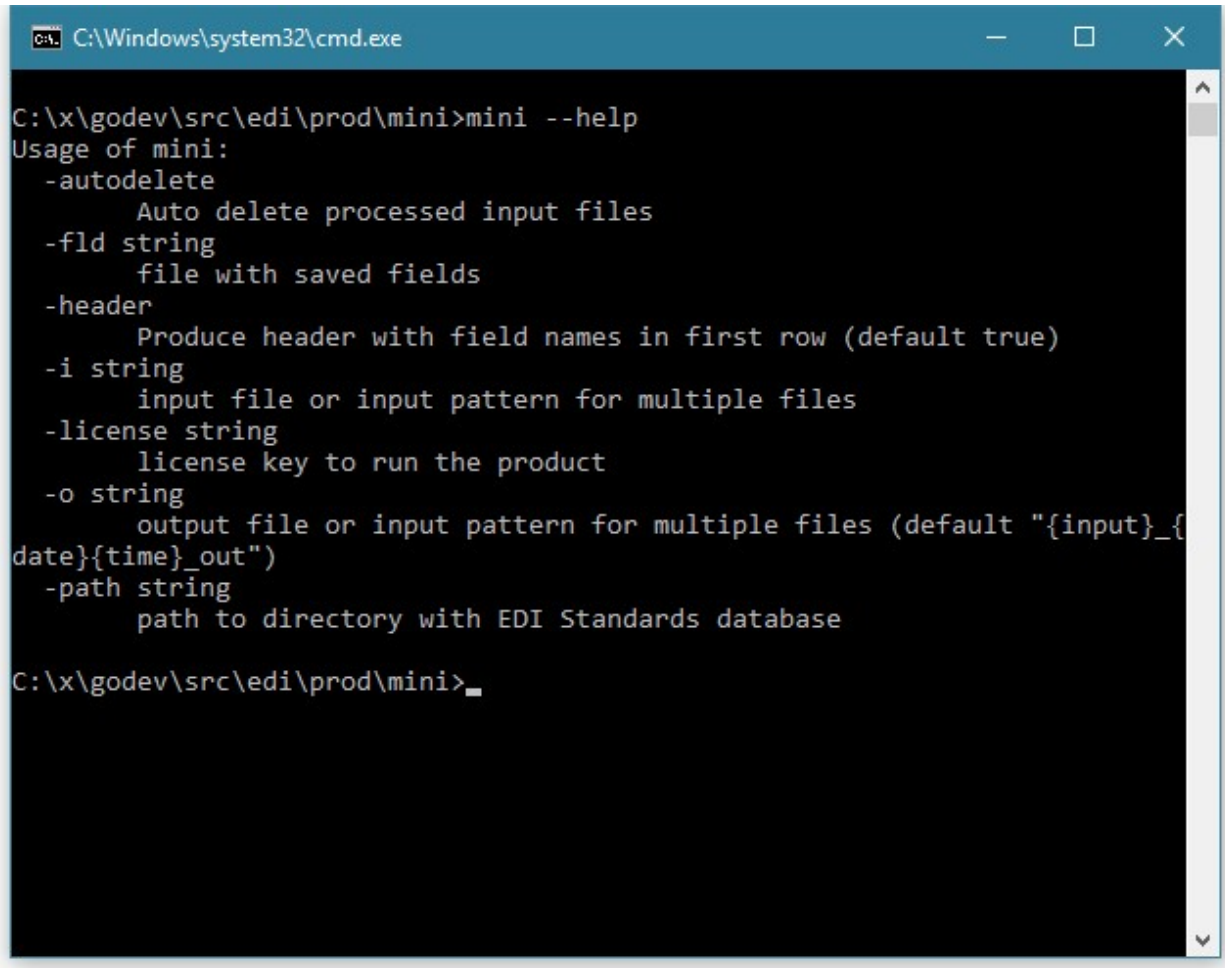
While GUI interface is more convenient command line interface lets you automate translations via scheduling tools on Windows and Linux. You can also run command line translation as part of your program (sometimes this is called ‘shell out’ execution). Mini Translator does not use any heavy runtimes such as JVM or .NET or any other runtimes that need JIT ‘warm-up’ for execution and might have slow startup.

Mini Translator ‘shell out’ is an option because load time for translator is small. It is designed to be fast.

Use utility called `mini` to run translations via command line. Command line tool has more flexible options. Run command line tool with `—help` parameter to find all available options.

Usually companies run automated jobs via command line. That means using scheduling tools or services to run jobs. Use Scheduled Tasks on Windows and `cron` on Linux if you do not have any other tools to automate processing.

Advantage of this approach is that you do not need to start GUI to run translations. Scheduled Tasks or `cron` will continue running it in the background even if GUI is closed (or never opened).

A screenshot of a Windows command prompt window. The title bar shows 'C:\Windows\system32\cmd.exe'. The command prompt is at 'C:\x\godev\src\edi\prod\mini>mini --help'. The output lists the usage and various options for the 'mini' command, including -autodelete, -fld, -header, -i, -license, -o, and -path. The prompt ends with 'C:\x\godev\src\edi\prod\mini>_'.

```
C:\Windows\system32\cmd.exe
C:\x\godev\src\edi\prod\mini>mini --help
Usage of mini:
  -autodelete
    Auto delete processed input files
  -fld string
    file with saved fields
  -header
    Produce header with field names in first row (default true)
  -i string
    input file or input pattern for multiple files
  -license string
    license key to run the product
  -o string
    output file or input pattern for multiple files (default "{input}_{
date}{time}_out")
  -path string
    path to directory with EDI Standards database
C:\x\godev\src\edi\prod\mini>_
```

Few important parameters:

Use “-i” parameter to specify input file. You can also use wildcards to process multiple files. For example:

```
mini -i=C:\prod\837\*.txt -license=EVGvtcG9yYXJ5fDIwMTYxMjA5fDF8RURJVkk=
```

This will process all files with *.txt extension in directory C:\prod\837.

Use “-o” parameter to specify output file name. By default output file name is formed using input file name, date, time, “_out” and specific extension (CSV).

You can setup various macros to form output file names. Current default is “{input}_{date}{time}_out”.

Available macros:

{input} – use input file name

{date} – use date

{time} – use time

{nano} – use nano seconds.

{nano} is handy when you have many very small files and want to make sure output name stays unique.

When processing files on Linux use “/” directory separator instead of “\”.

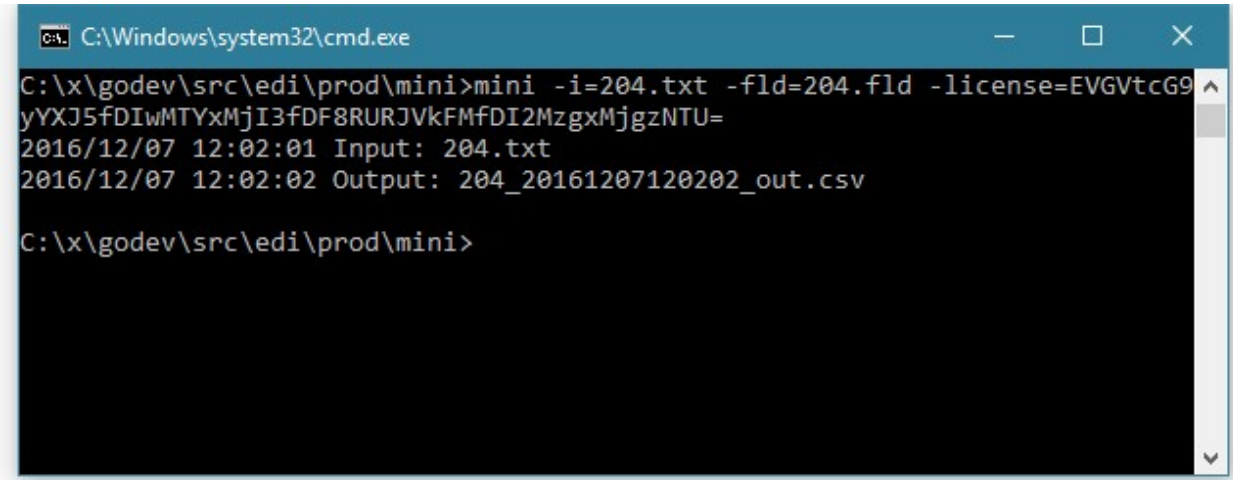
If you want to auto-delete input files when output file is produced pass `–autodelete=true` into command line utility.

If you do not want header with field names produced as first line of output pass `–header=false` into command line utility.

Example of basic use:

```
mini -i=204.txt -fld=204.fld -license=EVGvtcG9yYXJ5fDIwMTYxMjA5fDF8RURJVkFMfDMDQzDk=
```

This is very basic command line execution example. Our input file is called “204.txt”. We also pass 204.fld file. “204.fld” contains output field names added and saved via GUI.



```
C:\Windows\system32\cmd.exe
C:\x\godev\src\edi\prod\mini>mini -i=204.txt -fld=204.fld -license=EVGvtcG9yYXJ5fDIwMTYxMjA5fDF8RURJVkFMfDMDQzDk=
2016/12/07 12:02:01 Input: 204.txt
2016/12/07 12:02:02 Output: 204_20161207120202_out.csv
C:\x\godev\src\edi\prod\mini>
```

If you use `–fld` option and translator loads your previously saved fields it will use them on all input files that match input pattern. It is best to have only one type of transaction-release files in single directory. That way when you start translator with `–i=*.txt` option it will grab to process all files that are of the same transaction-release.

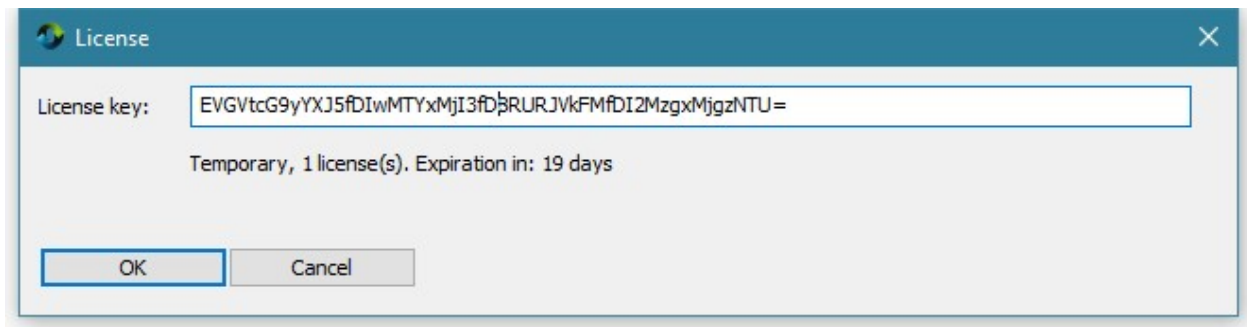
Same idea applies when using GUI interface. Once you load fields they will be used for all subsequent translations until you load new fields or clear already loaded fields using ‘New’ button.

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

License

Mini Translator comes with trial and subscription license. You can try and evaluate it free for 3 weeks. Once trail version expires please purchase subscription for continues 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.