

EDIDEV KNOWLEDGE LETTER
May 31, 2018
<http://www.edidev.com>

This is part two of a multipart article about managing and processing incoming EDI files from several trading partners. (The solution suggested here may not be suitable for all business scenarios.)

The Inbound Process - Part 2
=====

Now that you have received an EDI file in the inbound folder, your next step is to process it. But, how do you begin? If the EDI file content is unknown, what SEF file should be used to parse it? But to determine that, we need to know the content of the EDI file first. It's the chicken and the egg paradox. Well, one solution to this problem is to load all the SEF files for that trading partner that sent the EDI file, and then just let the FREDI component select the appropriate one.

To load several SEF files into an ediDocument object for TradingPartnerA, do the following:

```
ediDocument oEdiDoc = new ediDocument();

oEdiDoc.ImportSchema(sPath + "\TradingPartnerA\SEF\997_005010.SEF", 0);
oEdiDoc.ImportSchema(sPath + "\TradingPartnerA\SEF\850_005010.SEF", 0);
oEdiDoc.ImportSchema(sPath + "\TradingPartnerA\SEF\810_005010.SEF", 0);
etc...
```

(Just as a reminder of how we organized the SEF files and inbound EDI files, below is an illustration from our last article)

```
D:\App\HostCompany\

  +--\TradingPartnerA\
    +--\INBOUND\
    +--\SEF\

  +\TradingPartnerB\
    +--\INBOUND\
    +--\SEF\
```

In the above example, if the EDI file is an 850, or an 810 or even if it contains both the 850 and the 810, the ediDocument object would already have their schemas loaded in it, and therefore can process the EDI file.

KN20180501.txt

To process EDI files of another trading partner, you would do the same in a separate application or Windows Service. Basically, for each trading partner, you would have a separate application processing its own set of EDI files.

Windows Service A --> for processing TradingPartnerA EDI files
Windows Service B --> for processing TradingPartnerB EDI files

The advantage of having a separate application for each trading partner is that SEF files do not have to be loaded and unloaded each time there is a change of EDI files from many different trading partners. Also, EDI files of each trading partner can be processed concurrently. This will save processing time. But, the obvious disadvantage in this design is that it requires more RAM memory. You can minimize RAM usage by trimming the SEF files.

Normally, a trading partner's guideline will not use more than 20% of the data segments in a Transaction Set. This means that the other 80% is using up precious memory unnecessarily.

To trim a SEF file, first use the EDIdEv SEF Manager utility to edit it. Make sure you follow your trading partner's guideline, and actually delete (Ctrl-D) data segments that are not used. Then run it through our web utility SEF Trimmer inside your EDIdEv Customer Support website account (under License/Products page). The SEF trimmer will remove all data segments and elements in their respective directory that are not being used in the Transaction Set section of the SEF file. (As always, make a backup copy of your SEF file before making any modifications to it.)

Not only will a trimmed SEF file use less memory, but it will also load faster in FREDI!

More helpful hints in our next month's article - The Inbound Process - Part 3. (To be continued).

EDIDEV
"We make EDI fun!"