

# Non-XML Transformations Using Smooks

**Duration:** 5 Minutes.

This tutorial illustrates how Smooks can be used to transform non-XML character based data streams.

Internally, Smooks deals with data as a W3C DOM, parsing it to a DOM using SAX. Smooks allows you to target a SAX XMLReader implementation at a specific message in the same way it allows you target any other transformation resource. Using this, we can write a SAX parser to parse any character stream, converting it to a stream of SAX events. That parser can then be targeted at all relevant messages using a profile (*check out the other [tutorials](#) and [Smooks docs](#) for details on message targeting and profiling*).

## EDI as the Message Format

In this tutorial the message format will be an X12N EDI message as follows.

```
ISA*00*  *00*  *ZZ*EXTERNAL_TP  *ZZ*INTERNAL_TP
*010806*1200*U*00401*000000003*0*T* :
GS*HS*EXTERNAL_AP*INTERNAL_AP*20010101*120000*001*X*004010X092
ST*270*0001
BHT*0022*13*10001234*19990501*1319
HL*1**20*1
NM1*PR*2*ABC COMPANY*****PI*842610001
HL*2*1*21*1
NM1*1P*1*JONES*MARCUS*****SV*0202034
HL*3*2*22*0
TRN*1*93175-012547*9877281234
NM1*IL*1*SMITH*ROBERT*B***MI*11122333301
REF*1L*599119
DMG*D8*19430519*M
DTP*472*D8*19990501
EQ*98**FAM
SE*14*0001
GE*1*001
IEA*1*000000003
```

## Implementing the X12N to SAX Event Parser

So we need to implement an X12N to SAX Event Parser that will convert the above X12N stream into a stream of SAX events. This class is the [X12nToSaxEventParser](#).

X12nToSaxEventParser has 2 support class:

1. [X12nStreamReader](#): Wraps the stream and makes the X12N segments easier to access.
2. [X12nModel](#): Contains definitions to help the parser convert the X12N stream to a stream of SAX events.

## Targeting the X12N to SAX Event Parser

[x12n-config.cdrl](#) illustrates how to configure the X12nToSaxEventParser to parse messages originating from X12N based message producers ("x12n-requester" - "x12n-producer" might have been a better profile name).

## Sample Output

[sample-output.txt](#) show what the [X12nToSaxEventParser](#) produces and how Smooks "sees" the above X12N message.

## Conclusion

While the incoming message format may not be XML, as long as it's predictable, hierarchical etc, we can read and interpret the message as though it were an XML message. Once in XML, you can leverage the transformation and serialisation features of Smooks to generate whatever output is required.