

# Manual approach using code

# Automated approach with Flexter

## Analysis to understand structure of XML and XSD

1

For complex schemas this process can be super time consuming. The analysis can take more time than all the other steps combined.

- Go through documentation of industry data standard if available.
- Understanding XML Schema/Structure.
- Pinpoint the main elements, attributes, and their hierarchies within the XML.
- Understand how different elements and attributes relate to each other.
- Identify any hierarchical structures, such as parent-child relationships, within the XML data.
- Checking for Repeating and Optional Elements:
- Identify elements that occur multiple times and understand their significance.
- Note any optional elements that may or may not appear in the XML.
- Understanding Namespaces
- Evaluate the size of the XML files and the complexity of the structure, as this will impact the mapping process.
- Evaluate XML for Anomalies or Irregularities

## Model and create target schema

2

Once you have completed the analysis you need to use your findings to create a relational schema for the target format.

- Translate between the hierarchical structure to the relational format
- Map data types
- Define and implement naming conventions
- Define and implement parent-child relationships
- Define and implement constraints

1

## Analysis to understand structure of XML and XSD

Running a single command covers steps 1-3  
This creates a Flexter Data Flow, which analyses the XSD and/or XML, creates a relational target data model, and creates the mappings between XML source elements and target table columns

Option 1: Generating a data flow from an XML. We apply the elevate and reuse optimisation algorithm (g3 switch)

```
$ xml2er -g3 donut.xml
```

Option 2: Generating a Data Flow from an XSD

```
$ xsd2er -g3 donut.xsd
```

Option 3: Generating a Data Flow from an XML and XSD to get the best of both worlds

```
$ xml2er donut.xml
```

```
$ xsd2er -a4 -g3 donut.xsd
```

In this example we create the Data Flow from the command line. You can also use the Flexter API or Flexter UI to create a Data Flow.

2

## Model and create target schema

Already covered under Step 1

It is worthwhile mentioning that Flexter applies [optimisation algorithms](#) that significantly simplify the structure of the target schema

## Manual approach using code

### Create mapping

Manually map the XML elements to the target table columns

3

### XML Conversion

This is another very time consuming step.

- Developers write the code to load XML to the database and convert it with extensions to SQL such as XQuery to the relational database's tables and columns.
- Developers build a data pipeline and schedule a time to run it.
- The solution is tested

4

### Error handling, logging, and alerting

In XML mapping error handling, there are two crucial checks to perform. Firstly, ensure that the XML to be converted is valid. Secondly, verify that the XML adheres to the target schema. This means it should not have any additional XML elements and must comply with the specified data types.

5

## Automated approach with Flexter

### Create mapping

Already covered under Step 1

3

### XML Conversion

A single command is run to convert the data. The switch I3 references the ID of the Data Flow that was generated in the previous step.

```
$ xml2er -I3 donut.zip
```

In this example we create the XML Conversion from the command line. You can also use the Flexter API or Flexter UI to convert the XML.

4

### Error handling, logging, and alerting

This is automatically handled as part of Step 4 XML Conversion

5

## Manual approach using code

### Documentation

The most important documentation artefact is a Source to Target Map (STM). The STM acts as a blueprint that defines how each piece of data is transformed and transferred from the source to the target. It specifies how each field in the XML corresponds to the target including any data type conversions.

6

### Refactoring for changes

If changes are made to the XML or the XML schema you need to repeat steps 1-6 to cater for the change

7

## Automated approach with Flexter

### Documentation

Already covered under Step 1

When creating the Data Flow Flexter stores all of the information in its metadata catalog. The metadata can then be used to auto generate Source to Target Maps, ER diagrams, and diffs and deltas between different versions of a schema.

6

### Refactoring for changes

Flexter ships with a metadata store that semi-automates the upgrade between different versions of an XML or XSD.

7