

Title:	FX.Apps.XML – User Manual
Overview:	FX.Apps.XML is a namespace and package for applications which provide features to manipulate XML documents.
Purpose:	The purpose of this document is to describes and provides end user manual for applications under FX.Apps.XML package.

Table of Contents

Table of Contents	2
1 General.....	3
1.1 Content	3
1.2 Versioning	3
1.3 References.....	3
2 Applications	3
2.1 Test.....	3
2.2 Template	4
2.2.1 Content	4
2.2.2 Overview	4
2.2.3 Usage.....	7
2.2.4 Example	7

1 General

1.1 Content

Folder	Content
Apps	Executable applications
BuildLogs	Logs from build
Docs	Documentation
Libs	Output .Net libraries
Xsd	XSD schemes for e.g. configuration files.
EULA.txt	End user license agreements

1.2 Versioning

Versioning pattern of the package is:

<major>.<minor>.<release>.<revision>

where:

- <major> – Major number of the version. Version with different <major> numbers are incompatible.
- <minor> – Major number of the version. Version with different <major> numbers are incompatible.
- <release> – Release for target <major>.<minor> version. Version with the same <major>.<minor> numbers but greater <release> number has more features and is compatible with previous one. Compatible means that all data, all configuration all references which works for older version will work also for newer one.
- <revision> – Revision of the application. This number is increased every time when new code is committed. Version with the same <major>.<minor>.<revision> number has the same set of features and is compatible with previous one, but some bugs has been fixed.

1.3 References

Solution depends on FX.Core.NET packages. All documentation about logging configuration or other references can be found inside FX.Core.NET package.

2 Applications

2.1 Test

Location: Apps\Test

Description: Application is used for test and check if release has or hasn't bugs

2.2 Template

Location: Apps\Template

Description: Console Application which is used to manipulate XML structured documents. The base concept is to create output documents base on XML templates and csv data. This has been inspired to create cards using SVG standard but can be also used to any XML documents.

2.2.1 Content

Folder	Content
Apps\Template	Executable applications
Apps\Template\help.txt	Help
Apps\Template\log_config.xml	Configuration of logging
Apps\Template\license.xml	License for the application
Apps\Template\Logs	Default logs location
Libs	Output .Net libraries
Xsd\FX.App.XML.Lib.Template.XSD.Config.xsd	XSD schemes for your configuration files used to run the template.

2.2.2 Overview

The entire concept of the application has been shown on *Figure 1 – FX.App.XML.Template Design*. Application as an input requires XML files with file, page and elements templates (tokens) for output. All configuration how these templates should be processed is stored in configuration xml file. Additional input file is data csv file (Excel standard – semicolon separated with double quotas for string, double quotas for double quotas escape, LF line break inside string and CR LF line break). As an output application produces XML files (one for *Merged* mode and as many as pages for *File* mode).

Figure 2 – FX.App.XML.Template Algorithm shows general algorithm for application.

Figure 3 - Process Token Algorithm shows how target token (template) is processed and added to the output.

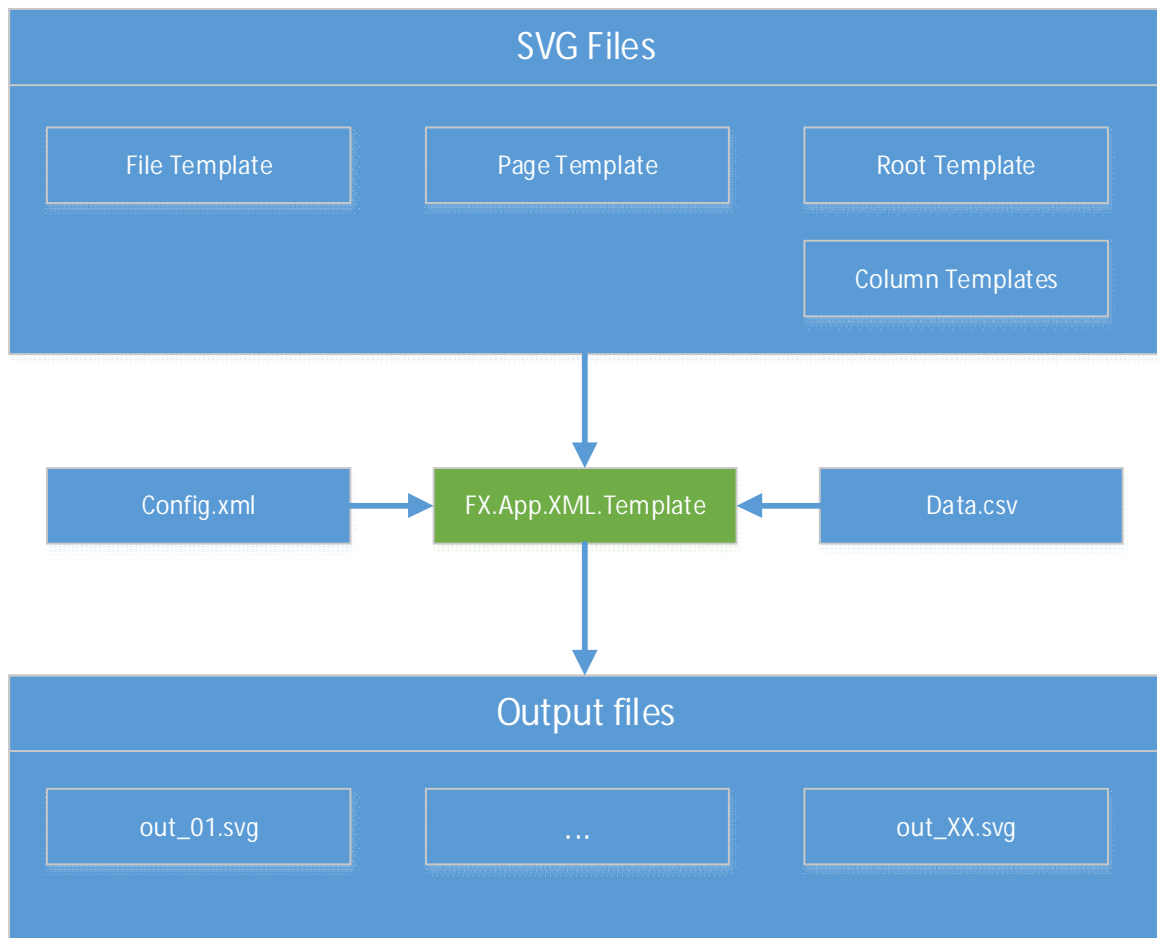


Figure 1 – FX.App.XML.Template Design

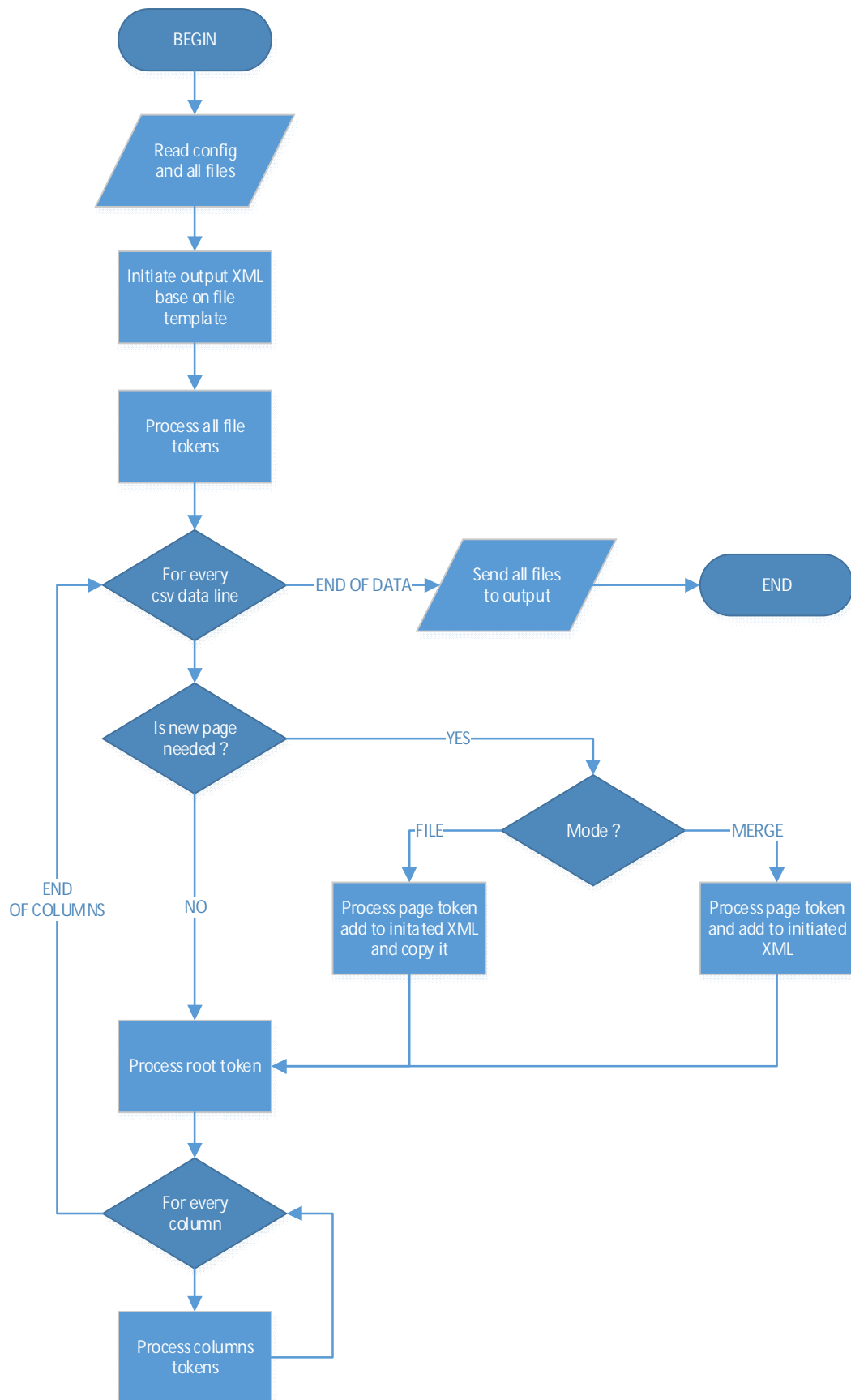


Figure 2 – FX.App.XML.Template Algorithm

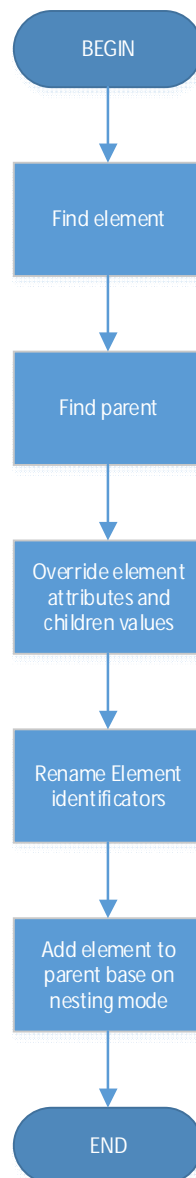


Figure 3 - Process Token Algorithm

2.2.3 Usage

Usage of the application can be found in help file: *Apps\Template\help.txt*.

2.2.4 Example

A good example for the usage of *FX.Apps.XML.Template* application is Test02 in location: *Apps\Test\Template\Test02*. For this example *Orgo* cards (created by Saul Pwanson - <http://www.pwanson.com/orgo/>) have been used. Original *Orgo* cards haven't been created by this application, example has been prepared based on reverse engineering.

Figure 4 - *Orgo* example shows overview of the example.

To run the example use the command:

```
..\..\..\Template\FX.App.XML.Template.exe -config config.xml -input data.csv -output Orgo\orgo_{0:00}.svg
```

From inside *Apps\Test\Template\Test02* location.

For logs please check *Apps\Template\Logs* location.

Just for note, *Orgo* uses *flowRoot* element for description rendering, so it will be not visible in browsers which don't support SVG 1.2 draft standard.

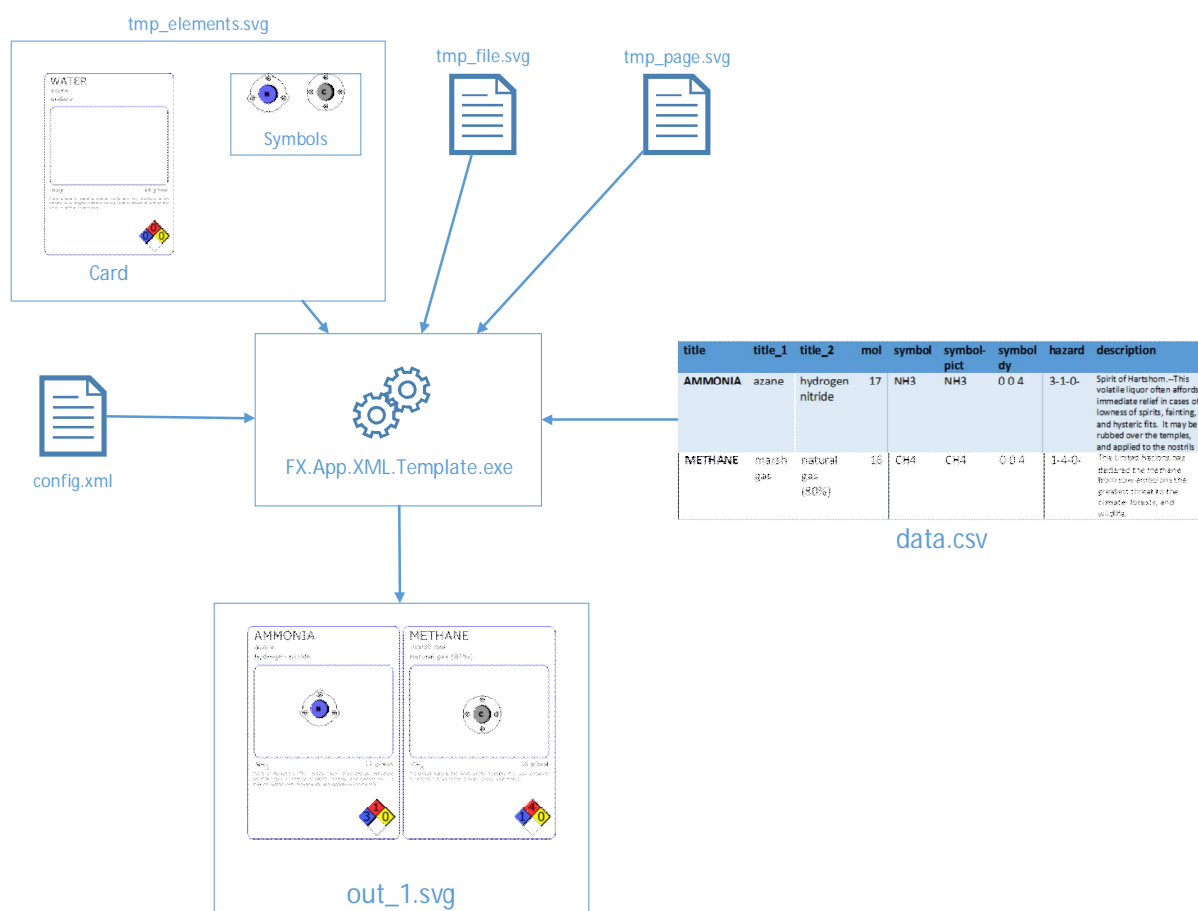


Figure 4 - Orgo example