

# LexisNexis Regulation Parser v.4.6.2

## *Source Code Documentation*

Author: Rebecca Ramnauth

Date: 18 November 2018

Namespace: [xml\\_converter](#)

Directory Tree:

Folder PATH listing for volume Windows

Volume serial number is 0AE0-C2BA

C:.

```
├── bin
│   ├── Debug
│   │   └── netcoreapp2.1
│   └── Release
│       └── netcoreapp2.1
├── obj
│   ├── Debug
│   │   └── netcoreapp2.1
│   └── Release
│       └── netcoreapp2.1
└── test_files
    └── inflow_content
        ├── 64632b65-fbc2-47ab-bf6c-136322eec66a
        ├── 8a12c008-8fdd-45e6-ab31-2499837a542f
        ├── 9703ad0c-ac31-4937-86fb-6b619ec847f9
        ├── 9a889fc7-5825-4062-b181-d432beb1b247
        ├── c528100b-a50e-4ffc-8c8b-f3ebbfef25e52
        ├── d11e89f1-0b5d-4879-bf01-6a0c16554639
        └── e8331a8f-4d07-4166-a1e6-f3e2ed4aaaae
```

## Class Index

[xml\\_converter.Program](#) reads and writes regulatory information

[xml\\_converter.Regulator](#) represents a regulatory body

[xml\\_converter.Regulation](#) represents the regulations that compose a regulatory body

[xml\\_converter.AppSettings](#) includes helper methods for using [ConfigurationManager](#) information

## App.config

*Startup version v4.0, .NETFramework v4.6.1*

### *appSettings*

#### Database Tables

**key** = "JobLog", Program static public attribute for corresponding job log table **job\_table**  
**key** = "FileLog", Program static public attribute for corresponding file log table **file\_table**  
**key** = "ContentLog", Program static public attribute for content log table **content\_table**

#### Inter-Table Relationships

**key** = "Reg\_Structure", a table defining the regulators and according tags to pull. The following levels are defined in individual tags:

**key** = "1", columns name in **Reg\_Structure** related to the file meta data  
 Precondition: last field/column must be a column that points to content meta column.  
**key** = "2", columns name in **Reg\_Structure** related to the content meta data  
 Precondition: first field must contain the content id, required.  
**key** = "3", columns name in **Reg\_Structure** defining user-defined (data not inferred/extracted from the raw input files) regulation items, e.g., the regulator/source name and subscription.  
 Precondition: first field must be subscription id, required.

**key** = "4", columns name in **Reg\_Structure** related to the actual content/regulation data  
**key** = "Structure-Log-Relationship", contains columns in ContentLog and their corresponding tag values from Reg\_Structure; each item is written in the following format:  
 column\_name, corresponding Reg\_Structure key/level, level index number;

#### Managing Directories

**key** = "TransferProcessed", Program private boolean for transferring parsed/processed files into key "ProcessedDir"  
**key** = "KeepXMLContent", Program private boolean for retaining EndDir directory

#### Directories

**key** = "StartDir", Program static public attribute for the inflow directory  
**key** = "EndDir", Program static public attribute for the XML content directory  
**key** = "ProcessedDir", Program static public attribute for processed raw files directory

### *connectionStrings*

**name** = "ConnectionKey"  
**connectionString** = "Data Source=[User];Initial Catalog=[Database];"  
**providerName** = "System.Data.SqlClient"

## Program Class Reference

`xml_converter.Program`

### Static Public Member Functions

`static void Main(string[] args)` contains parsing configures for hierarchical data, and makes calls for reading and writing files/directories

`static void ReadDir(String file, String to)` determines if reading nested directory or flat directory of raw input files, passing files 1-dimensionally to `ReadFile`

`static void ReadFile(String file, StreamWriter heads, String to)` contains parsing configurations for hierarchical data, and makes calls for reading and writing files/directories

`static void MoveDir(String file, String to)` moves a given directory/file into a given directory.

`static string GetTagNames(String file)`, a helper method for printing all tag names with corresponding namespace in a given file.

### Static Private Member Functions

`static SqlCommand CreateInsertCommand(string table, List<List<string>> relationship_configurations, Regulation regulation, SqlConnection connection)`, generates an appropriate SQL command based on relationship configurations defined in `App.config`

### Static Public Attributes

#### Directories

`static String start_folder`, Inflow from `ConfigurationManager` app key `"StartDir"`  
`static String end_folder`, Content from `ConfigurationManager` app key `"EndDir"`

#### SQL Data Client Settings

`static String connectionString`  
`static SqlConnection cnn = new SqlConnection(connectionString)`  
`static String job_table`, general job log  
`static String file_table`, regulation file log  
`static String new_content_table`, parsed regulation content table  
`static String new_structure_table`, tables structure and their relationships

## Regulator Class References

`xml_converter.Regulator`

### Public Member Functions

#### Constructor

`Regulator (string d, List<string> user_meta, List<string> file_meta, List<string> content_meta, List<string> reg_meta)`, a constructor requiring a regulator's container directory, and tags from the `new_structure_table` in `Program`

#### Printers

`void PrintNestedList (List<List<string>> s)` prints a 2-dimensional list of strings  
`void PrintList (List<string> s)` prints a 1-dimensional list of strings

#### Parsers

`List<Regulation> ParseReg (String file)` creates a list of the regulator's regulations by parsing by key, interpreting high-level meta data, and locating low-level meta data—in which each entry is then parsed as a `Regulation`

`List<string> InterpretTable (String file)` reads high-level meta data about a file,  
`App.config Reg_Structure level/key 1`

`List<List<string>> InterpretEntries (String file)` reads low-level meta data, about the content from listing/table of contents; `App.config Reg_Structure level/key 2`

`List<List<string>> InterpretRegDefinition ()` reads format configurations defined by the `Program` class, `App.config Reg_Structure level/key 4`, to determine which tags to pull

`List<string> Interpret (string key)` interprets which tag to pull by a key/configuration, dependent on the number of subkeys generated.

- `element[attribute=@value]`, pulls all tags called element with attribute matching value (3 subkeys are generated: element, attribute, and attribute's value)
- `element[attribute]`, gets first or default attribute value for tags called element (2 subkeys are generated: element and attribute)
- `element`, concatenates all inner text for tags called elements (1 subkey is generated: element)

#### Static Private Attributes

`string dir`, the main directory provided by the `Program` class. The regulator's directory is the concatenation of the `dir` and the regulator's subscription from `App.config Reg_Structure level/key 3`

`List<string> reg_definition`, the regulation keys, interpreted by `InterpretRegDefinition`

`List<string> file_definition`, the keys for file data, interpreted by `InterpretTable`

`List<string> content_definition`, the content data keys, interpreted by `InterpretEntries`

`List<string> user_content`, defined in `App.config Reg_Structure level/key 3`

## Regulation Class References

`xml_converter.Regulation`

### Public Member Functions

#### Constructors

`Regulation (string f, int n)`, a constructor requiring a file name, and the number of tags/columns to be parsed

#### Getters

`string GetFileItem (int idx)`, a getter method for accessing an element from file's meta  
`string GetContentItem (int idx)`, a getter method, accessing an element from content's meta  
`string GetUserItem (int idx)`, a getter method, accessing an element from user's meta  
`string GetRegulationItem (int idx)`, a getter method, accessing regulation item/field

#### Setters

`void AddFileItem (string item)`, appends an item to the end of file meta content  
`void AddContentItem (string item)`, appends an item to the end of content meta content  
`void AddUserItem (string item)`, appends an item to the end of user meta content  
`void SetRegulationItem (int idx, string item)`, inserts an item into regulation content

#### Parsers

`string ParseByKey (List<string> keys)` reads a regulation's corresponding .xml file and parses it by a specified key from a key of given keys. Given list of keys is determined at the `Regulator` level.

#### Static Private Attributes

```
string file
List<string> file_items
List<string> content_items
List<string> user_items
List<string> regulation_items
```

## AppSettings Class References

[xml\\_converter.Regulation](#)

### Public Member Functions

**static** T Get<T>(string key), casts key from [ConfigurationManager](#) into given type, use by stating [AppSettings.Get<dataType>\(key\)](#)