eliasen.eu software documentation

This document documents the software available at <http://www.eliasen.eu/DownloadSoftware.aspx>. You can find the newest version of this document at <http://www.eliasen.eu/Files/eliasen.eu.documentation.docx>

**Contents**

[1 Documentation of functoids 3](#_Toc221470463)

[1.1 Functoids 3](#_Toc221470464)

[1.1.1 Add Carriage Return and Line Feed 3](#_Toc221470465)

[1.1.2 Convert First Letter to Uppercase 4](#_Toc221470466)

[1.1.3 CSV Extract 5](#_Toc221470467)

[1.1.4 Date Converter 6](#_Toc221470468)

[1.1.5 New GUID 7](#_Toc221470469)

[1.1.6 Read Application Config 7](#_Toc221470470)

[1.1.7 String Replace 7](#_Toc221470471)

[1.2 Installation 8](#_Toc221470472)

[1.2.1 Installation for non-development PC 8](#_Toc221470473)

[1.2.2 Installation instructions for development PC's 8](#_Toc221470474)

[1.3 Known issues 8](#_Toc221470475)

[1.3.1 ID 8](#_Toc221470476)

[2 Documentation of pipeline components 9](#_Toc221470477)

[3 Documentation of winforms programs 9](#_Toc221470478)

[3.1 eliasen.eu – Draw 9](#_Toc221470479)

[3.1.1 Requirements 9](#_Toc221470480)

[3.1.2 Installation 9](#_Toc221470481)

[3.1.3 Usage 9](#_Toc221470482)

[3.1.4 Example 11](#_Toc221470483)

[3.1.5 Known issues 12](#_Toc221470484)

[3.1.6 To do 12](#_Toc221470485)

[3.2 eliasen.eu – ValidateXML 12](#_Toc221470486)

[3.2.1 Requirements 12](#_Toc221470487)

[3.2.2 Installation 13](#_Toc221470488)

[3.2.3 Usage 13](#_Toc221470489)

[3.2.4 Example 14](#_Toc221470490)

[3.2.5 Known issues 16](#_Toc221470491)

[3.2.6 To do 16](#_Toc221470492)

[4 License 16](#_Toc221470493)

[5 Warranty 17](#_Toc221470494)

[6 Help me 17](#_Toc221470495)

[6.1 Bugs, comments and feature requests 17](#_Toc221470496)

[6.2 Graphics 17](#_Toc221470497)

# Documentation of functoids

This section describes the implemented functoids.

## Functoids

### Add Carriage Return and Line Feed

This functoids is used in case you need to add either Carriage Return (CR), Line Feed (LF) or CRLF (both) to a string.

#### **Parameters**

The functoids takes four mandatory parameters:

1. The string to add the CR and/or LF to
2. A boolean determining whether to add the CR to the line
3. A boolean determining whether to add the LF to the line
4. A boolean determining whether the CR and/or LF should only be added in case they are not already present at the end of the line.

By setting the fourth parameter to “true”, it can be used to make sure that the CR and/or LF is always at the end of the string from the first parameter, but it will never appear twice.

If any of the last three inputs cannot be parsed as boolean types, a value of “false” is assumed.

#### **Functionality**

There is no way to add LFCR. The only three options are: CR, LF and CRLF. This is because they are the only three options I have seen in use.

The functionality of the functoid is given here in pseudo code:

* Create empty string called <crlfstring>
* If second parameter is true, add CR to <crlfstring>
* If third parameter is true, add LF to <crlfstring>
* If fourth parameter is:
  1. true
     1. If the first parameter has <crlfstring> at the end, return the first parameter
     2. If the first parameter does not have <crlfstring> at the end, add <crlfstring> to the first parameter and return the result of this concatenation
  2. false
     1. Add <crlfstring> to the first parameter and return the result of this concatenation

#### **Examples**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **First** | **Second** | **Third** | **Fourth** | **Result** |
| Jan | true | true | true | JanCRLF |
| Jan | true | true | false | JanCRLF |
| Jan | true | false | true | JanCR |
| Jan | true | false | false | JanCR |
| Jan | false | true | true | JanLF |
| Jan | false | true | false | JanLF |
| Jan | false | false | true | Jan |
| Jan | false | false | false | Jan |
| JanCR | true | true | true | JanCRCRLF |
| JanCR | true | true | false | JanCRCRLF |
| JanCR | true | false | true | JanCR |
| JanCR | true | false | false | JanCRCR |
| JanCR | false | true | true | JanCRLF |
| JanCR | false | true | false | JanCRLF |
| JanCR | false | false | true | JanCR |
| JanCR | false | false | false | JanCR |
| JanLF | true | true | true | JanLFCRLF |
| JanLF | true | true | false | JanLFCRLF |
| JanLF | true | false | true | JanLFCR |
| JanLF | true | false | false | JanLFCR |
| JanLF | false | true | true | JanLF |
| JanLF | false | true | false | JanLFLF |
| JanLF | false | false | true | JanLF |
| JanLF | false | false | false | JanLF |
| JanCRLF | true | true | true | JanCRLF |
| JanCRLF | true | true | false | JanCRLFCRLF |
| JanCRLF | true | false | true | JanCRLFCR |
| JanCRLF | true | false | false | JanCRLFCR |
| JanCRLF | false | true | true | JanCRLF |
| JanCRLF | false | true | false | JanCRLFLF |
| JanCRLF | false | false | true | JanCRLF |
| JanCRLF | false | false | false | JanCRLF |

### Convert First Letter to Uppercase

This functoids is used to convert strings from systems that supply strings in either uppercase or lowercase into “normal” strings. This means, that only the first character of each substring in the input is returned as uppercase and the rest fo the string is returned as lowercase.

#### **Parameters**

The functoids takes two parameters, of which the second is optional.

1. The input to convert
2. A comma separated list of strings that describes the strings that the functoid should leave exactly as described in this string.

The second parameter is optional. If it is not present, all substrings of the input are converted to have a capital first letter.

#### **Functionality**

The functionality of the functoid is given here in pseudo code:

1. The string in the first input is split into an array of strings using a space as the separator.
2. For each string in the array, check if it is also present in the list of strings from the second parameter.
3. If the string form parameter 1 is present in the list of strings in parameter 2, then the string from parameter 2 is added to the output.
4. If the string isn’t present, it is converted to lowercase and the first letter is capitalized. Then the string is added to the output

#### **Examples**

|  |  |  |
| --- | --- | --- |
| **First** | **Second** | **Result** |
| jan eliasen |  | Jan Eliasen |
| jan eliasen A/S | Jan | JAn eliasen A/s |
| JAN ELIASEN A/S | Jan,A/S | Jan Eliasen A/S |
| This is a test string for the functoid | tHIS,A,fOr,silly | tHIS Is A Test String fOr the functoid |

#### Note

Please note, that you can use the functoid described in section 1.1.6 to get a comma separated list of strings at runtime form the app.config belonging to BizTalk.

### CSV Extract

This functoids is used to retrieve a specific substring of a string separated by some value.

#### **Parameters**

The functoids takes three mandatory parameters:

1. The input to split
2. The character(s) that is used to split the string into a list of substrings
3. The position of the substring in the list of substrings that you want as output

The second parameter can consist of more than one character. In that case, the characters should just be written as a string, no separator.

If the input string is split up into fewer substrings than the third parameter indicates you want returned, then an empty string is returned. If the third character cannot be parsed as an integer, a 0 is assumed. The third parameter specifies the index into a .NET array, meaning that it is zero-based, so it starts at 0 and ends at the length of the array minus 1.

#### **Functionality**

The functionality of the functoid is given here in pseudo code:

1. The string in the second parameter is converted into an array of characters.
2. The string given in parameter one is split into an array of substrings, using the array of characters. The method used is the standard .NET 2.0 String.Split method.
3. The third parameter is converted to an integer and used as an index into the array of substrings, and the resulting string is returned.

If the third parameter cannot be parsed as an integer, a 0 is assumed. If the parsed integer is greater than the number of substrings, an empty string is returned.

#### **Examples**

|  |  |  |  |
| --- | --- | --- | --- |
| **First** | **Second** | **Third** | **Result** |
| 12,34,56 | , | 0 | 12 |
| 12,34,56 | , | 1 | 34 |
| 12,34,56 | , | 6 |  |
| 12,34,56 | , | a | 12 |
| 12,34,56 | ,3 | 0 | 12 |
| 12,34,56 | ,3 | 1 |  |
| 12,34,56 | ,3 | 2 | 4 |
| 12,34,56 | ,3 | 3 | 56 |

### Date Converter

This functoids is used to convert between different date and time formats.

#### **Parameters**

The functoids takes three mandatory parameters:

1. The input date or datetime
2. A string describing the input format of the first parameter
3. A string describing the format out the output

The strings for parameters two and three are the standard .NET 2.0 date formats, which can be seen at <http://msdn.microsoft.com/en-us/library/az4se3k1(VS.80).aspx>. The string “s” represents the standard ISO 8601 datetime format (yyyy-MM-ddTHH:mm:ss) which is also the standard datetime format for XML, where the data type datetime is used. So the “s” will probably come in handy.

#### **Functionality**

The functionality of the functoid is given here in pseudo code:

1. Parse the first parameter using the format from the second parameter
2. If parsing went well, output datetime using format given in parameter three

If any exceptions occur during the execution of this functoid, the input is returned unaltered.

#### **Examples**

|  |  |  |  |
| --- | --- | --- | --- |
| **First** | **Second** | **Third** | **Result** |
| 2008-10-20T10:20:30 | s | r | Mon, 20 Oct 2008 10:20:30 GMT |
| 2007-09-19T12:34:56 | s | d | 19-09-2007 |

### New GUID

This functoids simply returns a newly created unique GUID, which can be used if you need a unique identifier for a row to insert into a database, an identifier to correlate on inside BizTalk or other purposes.

#### **Parameters**

The functoids takes no parameters

#### **Functionality**

The functoids uses the .NET framework to generate a new and uniue GUID, which is returned.

#### **Examples**

|  |  |
| --- | --- |
| Input | Output |
| N/A | A7e234f4-f790-4f9a-a097-db000b50e2d7 |

### Read Application Config

This functoids is used to read a specific value from the BTSNTSvc.exe.config configuration file at runtime. This way you can have configuration settings needed in a map in this configuration file so you don’t have to either use the DB Lookup functoid or have the ettings somewhere else, and load them into an orchestration and merge two XML documents.

#### **Parameters**

The functoids takes one mandatory parameter:

1. The key of the application setting to retrieve.

#### **Functionality**

The functoid simply uses the .NET 2.0 System.Configuration.ConfigurationSettings.AppSettings[key] method to retrieve the needed value. Yes, I know it is deprecated, but it works ☺

#### **Examples**

|  |  |  |
| --- | --- | --- |
| **app.config** | **First** | **Result** |
| <appSettings>  <add key="eliasen" value="eliasenValue"/>  </appSettings> | eliasen | eliasenValue |
| <appSettings>  <add key="eliasen" value="eliasenValue"/>  </appSettings> | nesaile |  |

### String Replace

This functoids is used to replace all occurrences of a given substring inside a string with another string.

#### **Parameters**

The functoids takes three mandatory parameters:

1. The string to that needs substrings replaced
2. The string to search for and replace
3. The string to insert into the first parameter instead of all occurrences of the second parameter

#### **Functionality**

The functoid simply uses the .NET 2.0 String.Replace method, passing the parameters straight to this method, and returning the result from this method call.

The functionality of the functoids is given here in pseudocode:

1. Return param1.Replace(param2, param3)

#### **Examples**

|  |  |  |  |
| --- | --- | --- | --- |
| **First** | **Second** | **Third** | **Result** |
| Jan | a | b | Jbn |
| Jan Eliasen | n | yeah | Jayeah Eliaseyeah |

## Installation

### Installation for non-development PC

1. Unzip the zip file
2. Put the DLL in the GAC, using your favorite way of doing that.
3. Optional: Delete old version of assembly from GAC. Do NOT do this, if you have running solutions that are not updated to reference the new version yet.

### Installation instructions for development PC's

1. Follow the two above steps for non-development PC
2. Copy the dll file to the "%BTS%\Developer Tools\Mapping Extensions" folder, where %BTS% is the installation folder of BizTalk.
3. Start Visual Studio.NET 2005 and go to "Tools" => "Choose Toolbox Items" and then go to the "Functoids" pane. Check the functoids that should be available in the toolbox.
4. Alternatively, right click on the toolbox and choose "Reset Toolbox".

If you encounter difficulties refreshing the toolbox, you may be running into an issue with Visual Studio .NET 2005, which I have described [here](http://blog.eliasen.dk/2006/12/05/RemovingFunctoidFromToolbox.aspx). Note the comments for the post, as this is the easiest way to solve the issue.

## Known issues

### ID

All functoids must have a unique ID in order for the runtime to distinguish them. The ID is an integer and all IDs below 6000 are reserved for Microsoft use. The functoids in the eliasen.eu.BizTalk.Functoid library have the following IDs:

|  |  |
| --- | --- |
| **Functoid name** | **ID** |
| Add Carriage Return and Line Feed | 6424 |
| CSV Extract | 6428 |
| Date Converter | 6427 |
| New GUID | 6425 |
| String Replace | 6426 |

In case you run into an ID duplicate which you need fixed, please contact (<http://www.eliasen.eu/Contact.aspx>) me and I will provide you with a new library with different IDs.

# Documentation of pipeline components

No pipeline components available yet.

# Documentation of winforms programs

This section describes the available winforms programs.

## eliasen.eu – Draw

This section describes the “eliasen.eu – Draw” program.

### Requirements

To install and run this program, you need the following:

* Computer running Windows XP or later
* .NET Framework 3.5 SP1
* Windows Installer 3.1

### Installation

1. Unzip the installer
2. Run the installer
3. Follow the wizard

### Usage

This utility is a small Winforms program, that can be used to make a draw between several contestants. It can be used to draw between a number of constants with each one chance of winning, but also to draw between a number of contestants with diverse number of chances. So for instance, if the constants are supposed to get a chance of winning in the draw relecting the number of beers she has been drinking, one person might have 10 chances and another person mght have 15 chances, corresponding to 40% and 60% chance of winning respectively.

The main window can be seen in Figure 1.

It has a menu which consists of:

* + - * “File”
        1. Exit
      * “Help”
        1. About

Beneath the menu, a grid with all entered contestants can be seen. The grid has four columns:

1. Name – The name of the constant.
2. Points – the number of chances the contestant has.
3. Winning Chance – The chance the contestant ahs of winning the draw. It is basically the number of points divided by the total number of points
4. Test chance – If you perform a test on the drawing to confirm that the program is actually random, this collumn will show the percentage of the testruns where this contestant won.

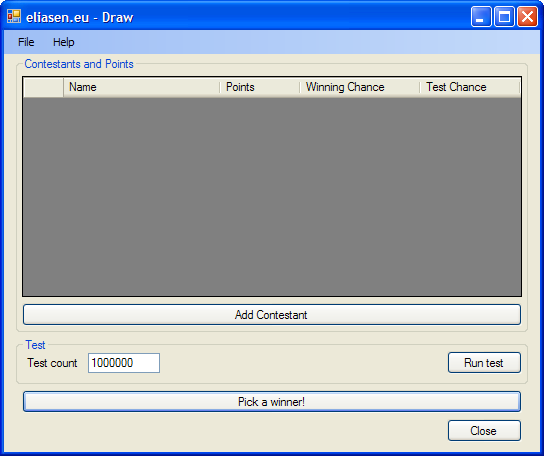


Figure : Main Window of eliasen.eu – Draw

The “Add Contestant” button opens up a dialog, where a new conestant can be entered, This can be seen in Figure 2.

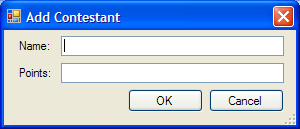


Figure : Add Contestant dialog

The “Test” group of the main window contains the test functionality. You can enter how many times the test run should draw a winner and then click on “Run test”. The grid will be updated to reflect the number of wins by each contestant.

The “Pick a winner!” button draws a winner and outputs the name in a message box.

The “Close” button closes the program.

### Example

Figure 3 shows an example run of the program. As you can see, three contestants have been added with different number of points. The test run has done one million drawings just to see the outcome, and the percentages in the “Winning Chance” and the “Test Chance” columns are close enough to be correct.

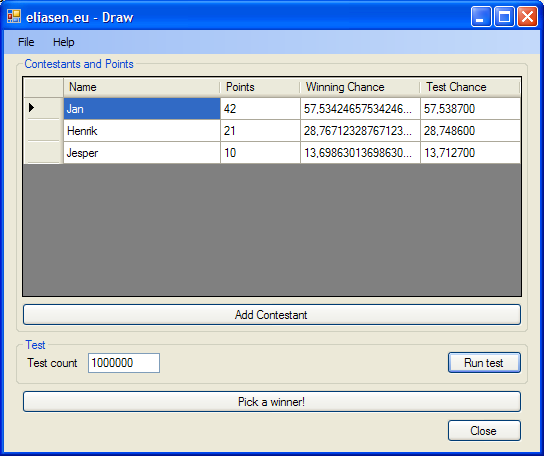


Figure : Example run of eliasen.eu - Draw

### Known issues

None

### To do

1. Add an icon to the program
2. Add help to the program
3. Add tooltips
4. Add localization to support multiple languages

## eliasen.eu – ValidateXML

This section describes the “eliasen.eu – Draw” program.

### Requirements

To install and run this program, you need the following:

* Computer running Windows XP or later
* .NET Framework 3.5 SP1
* Windows Installer 3.1

### Installation

1. Unzip the installer
2. Run the installer
3. Follow the wizard

### Usage

The main window can be seen in Figure 4.

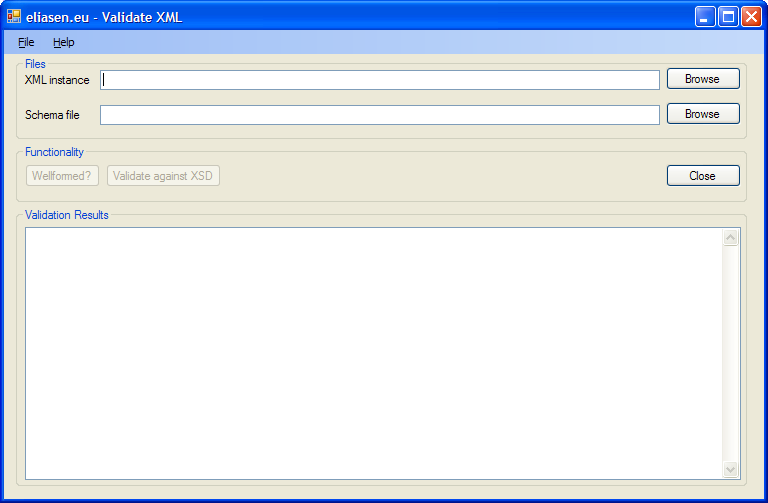


Figure : Main Window for ValidateXML

It has a menu which consists of:

* + - * “File”
        1. Open XML Instance
        2. Open XSD Schema
        3. Exit
      * “Help”
        1. About

It has the possibility to browse for both an XML file and an XSD schema. If the textbox for the XML file has a valid file name, the “Welformed?” button gets enabled, allowing you to check the file to see if it is wellformed XML. If the textbox for the XML file has a valid file name and the textbox for the schema has a valid file name, then the button “Validate against XSD” gets enabled, allowing you to validate the XML file to check that its structure complies to the schema.

When validating against a schema, the check for wellformedness is performed first, and the validation against the schema is only performed if there were no errors checking for wellformedness.

The large textbox at the bottom will contain any errors that might appear.

### Example

This section contains three screenshots of the program in action. The figure captions contain descriptions of the screenshots.

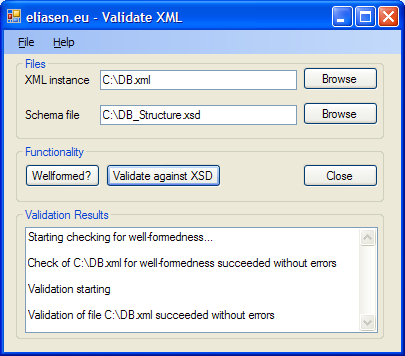


Figure : Example of validation of XML against XSD, without any errors

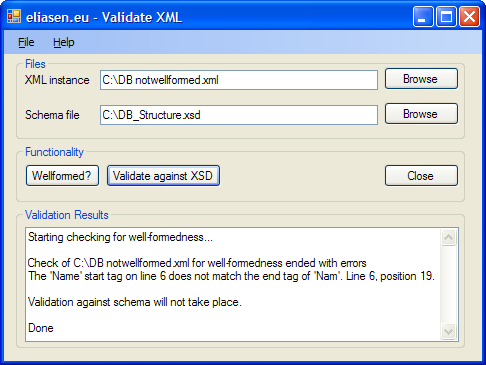


Figure : Example of validation of XML against XSD, where the XML isn't well formed. No validation against the schema is performed, and a descriptive error message is provided.

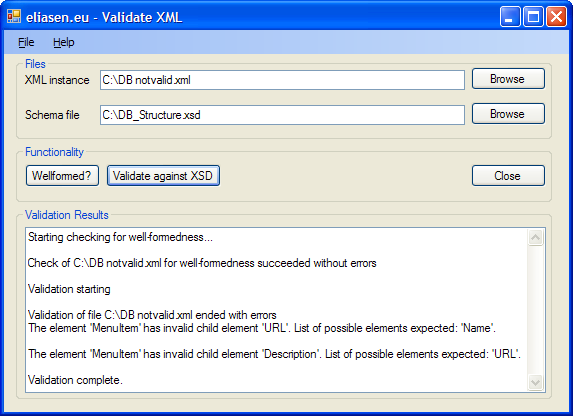


Figure : Example of validation of XML against XSD, where the XML does not conform to the XSD. Two errors are discovered and described to the user.

### Known issues

None

### To do

1. Support DTDs
2. Support XDRs
3. Add an icon to the program
4. Add help to the program
5. Add tooltips
6. Add localization to support multiple languages
7. Add the possibility to email the validating results

# License

All the software that can be downloaded here is "optional-postcardware", meaning, that I would really appreciate a postcard from you if you use my software, just so I know how many people use it :-) The postcard can be a paper postcard sent by ordinary mail or it can be an email. In any ase, find my contact information at <http://www.eliasen.eu/Contact.aspx>. Note, the postcard is optional. You are not breaking any license terms by not sending me a postcard :-)

# Warranty

Basically, any use of these products is your own responsibility. I provide no warranty, promises or anything else, and I will not pay you for lost work, earnings, etc.

# Help me

## Bugs, comments and feature requests

Should you encounter any bugs at all, please let me know. Also, any comments and requests for features are welcome. Find my contact information at <http://www.eliasen.eu/Contact.aspx>. Thanks.

## Graphics

As you can probably see from my extremely lousy icons, layouts, and so on, graphics really isn't one of my strong sides. So if you are good at creating icons and so on, and would like to help me with this part, please conact me via <http://www.eliasen.eu/Contact.aspx>