



XML Utility SL

The library "XML Utility SL" contains function blocks to read and write xml files or xml strings on a control.

Product description

The library "XML Utility SL" contains function blocks to read and write xml files or xml strings on a control. All elements of the xml file are stored in an array of structures. The library also contains function blocks to find elements by name and attribute. The usage of the library is demonstrated by two example applications.

The library can read and write UTF-8 and UTF-16 coded well-formed xml files.

Supported xml elements:

- Standard element `<xxx>aaa</xxx>` and `<xxx />`
- Attributes `yyy="zzz"`
- Comments `<!-- Comment -->`
- CDATA elements

Each element of the file is stored in a structure of the type `XMLElement`. A complete xml file or parts of it will be stored in an array of structures.

Fields of the structure `XMLElement`:

`diParentIndex`: Position of the parent element in the data array

`wsName`: Name of the element

`wsValue`: Value of the element

`udiPosition`: Position of the element in the file

`elementType`: Type of the element (Element, Attribute, CDATA-Element, Comment)

Function block to read xml files

`XMLGetElement`:

Function block to read xml elements with following options:

- Read one element only
- Read all children of an element
- Read all underlying elements

The reading is done file based and buffered, so it is possible to load an element without loading the complete file.

`XMLGetElementAsync`:

Asynchronous version of the function block XMLGetElement.

Function blocks to find xml elements

XMLFindElement:

Function block to find xml elements with following search options:

- Search by element name
- Search by attribute name and value
- Read all children of the search result
- Read all underlying elements of the search result
- Start position in the file of the search

XMLFindElementAsync:

Asynchronous version of the function block XMLFindElementAsync.

XMLFindElementByStringAsync:

Functions like function block XMLFindElementAsync with STRING input fields.

Function blocks to write xml files

XMLWrite:

Function block to write xml files. Partial writing of elements is not supported, i.e. the complete file structure must be loaded.

XMLWriteAsync: Asynchronous version of the function block XMLWrite.

Parameter

Reading files is buffered. The size of the buffer must be greater than the size of the largest xml element. The size of the buffer and the maximum WSTRING size of the structure XMLElement can be changed in the parameter list of the library manager (see figure 1). The size of the data array can be declared individually outside the function blocks.

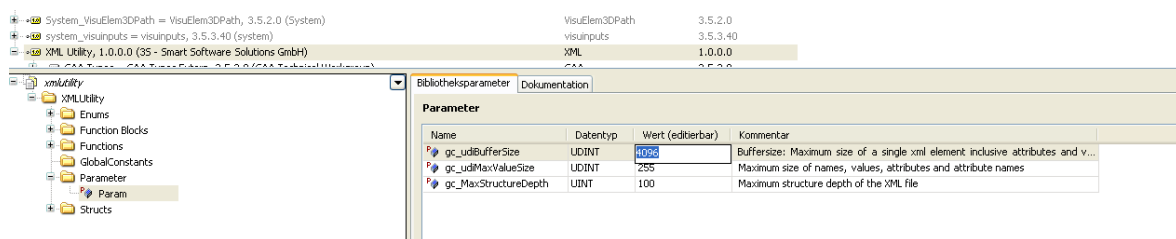


Figure 1: Parameter list of the xml library

Example project XML Utility SL Example.project

The example project XML Utility SL Example contains two example applications for the xml library. The application XMLFindExample shows how to search for elements and how to load child elements (see figure 2). The application has three functions. The button “Find first XML element” returns the first element with the specified element name and attribute value. If no element name is set the root element will be returned. The button “Find next element” will

continue the search from the current position and will return the next element. The function "Read children of the selected element" reads all child elements of the selected element. You can use the example XML file "SimpleXMLEExample.xml" to test the application.

Note

To test the examples on a CODESYS Control Win the files "SimpleXMLEExample.xml" and "SimpleXMLEExample_UTF16.xml" can be copied from the target directory of the installation into the directory "c:\temp".

Example “SimpleXMLExample.xml”:

```
<?xml version="1.0" encoding="utf-8"?>
<Elements>
  <!--This is a simple xml example -->
  <Element attribute1="1">
    This is the value of Element 1
  </Element>
  <Element attribute1="2">
    This is the value of Element 2
  </Element>
  <Element attribute1="3" attribute2="33789" attribute3="99.98">
    This is the value of Element 2
  </Element>
  <Element attribute1="4">
    <Subelement id="1">
      Value of Subelement 1
    </Subelement>
    <Subelement id="2">
      Value of Subelement 2
    </Subelement>
    <Subelement id="3">
      Value of Subelement 3
      <!--3. level -->
      <Child attribute1="1">
        Element4/Subelement3/Child1
      </Child>
      <Child attribute1="2">
        Element4/Subelement3/Child2
      </Child>
      <Child attribute1="3">
        Element4/Subelement3/Child3
      </Child>
    </Subelement>
  </Element>
  <!--CDATA and escaping -->
  <Text id="1">
    <![CDATA[This is a CDATA section.]]>
  </Text>
  <Text id="2">
    Lesser than: &lt; Greater than: &gt; And: &amp; Apostroph: &apos; Quote:
  </Text>
</Elements>
```

The application XMLReadWriteExampleAsync reads and writes xml files (see figure 3 and 4). The function “Read XML file” loads every element of the file “SimpleXMLExample.xml” in an array of XMLElements. The function “Write XML file” writes the loaded elements in a new file called “SimpleXMLExample2.xml”. Each element of the file is loaded to the memory. It is important to ensure that the size of data array is large enough to hold the data.

General information

Supplier:

CODESYS GmbH
 Memminger Strasse 151
 87439 Kempten
 Germany

Support:

Technical support is not included with this product. To receive technical support, please purchase a CODESYS Support Ticket.

<https://support.codesys.com>

Item:

XML Utility SL

Item number:

Sales/Source of supply:

CODESYS Store
<https://store.codesys.com>

Included in delivery:

CODESYS Package with example project

System requirements and restrictions

Programming System	CODESYS Development System V3.5.15.0
Runtime System	CODESYS Control V3.5.15.0
Supported Platforms/ Devices	All
	Note: Use the project "Device Reader" to find out the supported features of your device. "Device Reader" is available for free in the CODESYS Store.
Additional Requirements	-
Restrictions	-
Licensing	License activation optional on CODESYS Key or Soft Key (Soft Key: free of charge component of CODESYS Controls) Licensing via Soft Key is strictly linked to hardware. Note: Without a license the software runs for 30 minutes in demo mode.
Required accessories	-

Note: Technical specifications are subject to change. Errors and omissions excepted. The content of the current online version of this document applies.

Creation date: 2023-08-21