Knihovna XmlLib

TXV 003 63.01
první vydání
prosinec 2010
změny vyhrazeny

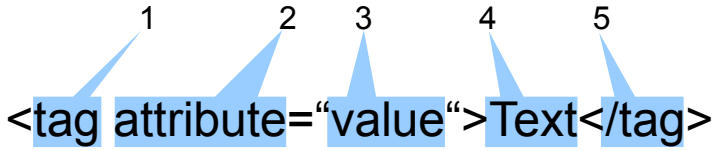
Historie změn

Datum	Vydání	Popis změn
Prosinec 2010	1	První vydání, popis odpovídá XmlLib_v13

OBSAH

1 ÚVOD

Knihovna XmlLib je standardně dodávaná jako součást programovacího prostředí Mosaic. Knihovna obsahuje funkční bloky umožňující práci s daty ve formátu Extensible Markup Language (XML).



Obecná struktura XML

1 – tag
2 – atribut
3 – hodnota atributu
4 – text
5 – ukončení tagu

XmlLib ve verzi 1.3 nepokrývá kompletně všechny možnosti zápisu XML dat. Omezení knihovny jsou následující:

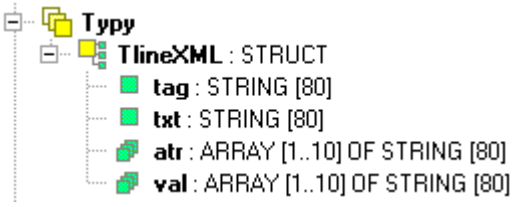
- Komentáře jsou ignorovány
- Maximální délka názvu tagu, atributu nebo obsahu tagu či atributu je 80 znaků
- Maximální počet atributů pro jeden tag je deset
- Není podporována sekce CDATA
- Nejsou podporovány zástupné xml entity

Následují obrázek ukazuje strukturu knihovny XmlLib v prostředí Mosaic

Pokud chceme funkce z knihovny XmlLib použít v aplikačním programu PLC, je třeba nejprve přidat tuto knihovnu do projektu. Knihovna je dodávaná jako součást instalace prostředí Mosaic od verze v2.0.25. Knihovna je závislá na knihovně SysLib.

2 DATOVÉ TYPY

V knihovně XmlLib jsou definovány následující datové typy:

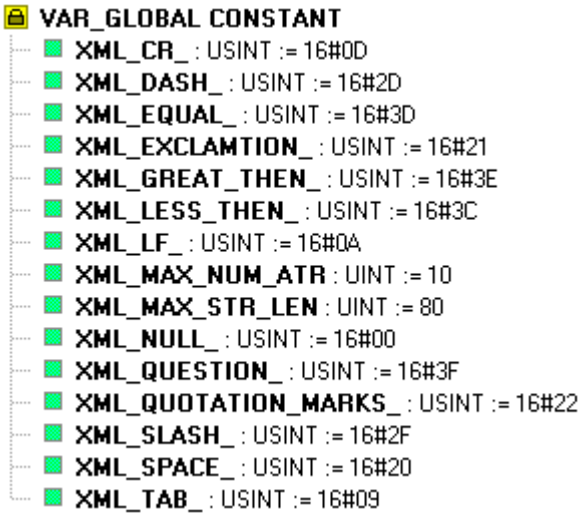


Datový typ *TlineXML* je struktura obsahující informace o aktuálně načteném elementu XML dokumentu. Význam jednotlivých položek je následující:

Identifikátor	***Typ***	***Význam***
TlineXML	STRUCT	Popis elementu XML
.tag	STRING	Jméno tagu včetně klíčových znaků ?, /, atd.
.txt	STRING	Text před vlastním tagem
.atr	ARRAY [1..10] OF STRING	Názvy prvních deseti atributů
.val	ARRAY [1..10] OF STRING	Hodnoty prvních deseti atributů

3 KONSTANTY

V knihovně XmlLib jsou definovány následující konstanty:



Konstanty se využívají při zpracování XML dokumentu. Význam konstant je následující:

Identifikátor	*Typ*	*Hodnota*	*Význam*
XML_NULL_	USINT	16#00	konec řetězce
XML_TAB_	USINT	16#09	tabelátor
XML_LF_	USINT	16#0A	novy řádek
XML_CR_	USINT	16#0D	konec řádku
XML_SPACE_	USINT	16#20	' ' znak 'mezera'
XML_EXCLAMTION_	USINT	16#21	'!' znak 'vykřičník'
XML_QUOTATION_MARKS_	USINT	16#22	'"' znak 'uvozovky'
XML_DASH_	USINT	16#2D	'-' znak 'pomlčka'
XML_SLASH_	USINT	16#2F	'/' znak 'lomítko'
XML_LESS_THEN_	USINT	16#3C	'<' znak 'menší než'
XML_EQUAL_	USINT	16#3D	'=' znak 'rovná se'
XML_GREAT_THEN_	USINT	16#3E	'>' znak 'větší než'
XML_QUESTION_	USINT	16#3F	'?' znak 'otazník'
XML_MAX_STR_LEN	UINT	80	max. velikost řetězce tagu/textu/atributu/hodnoty v XML
XML_MAX_NUM_ATR	UINT	10	max. počet atributu v XML řádků

4 GLOBÁLNÍ PROMĚNNÉ

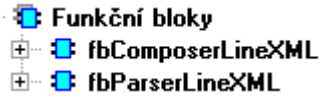
V knihovně XmlLib nejsou definovány žádné globální proměnné.

5 FUNKCE

V knihovně XmlLib nejsou definovány žádné funkce.

6 FUNKČNÍ BLOKY

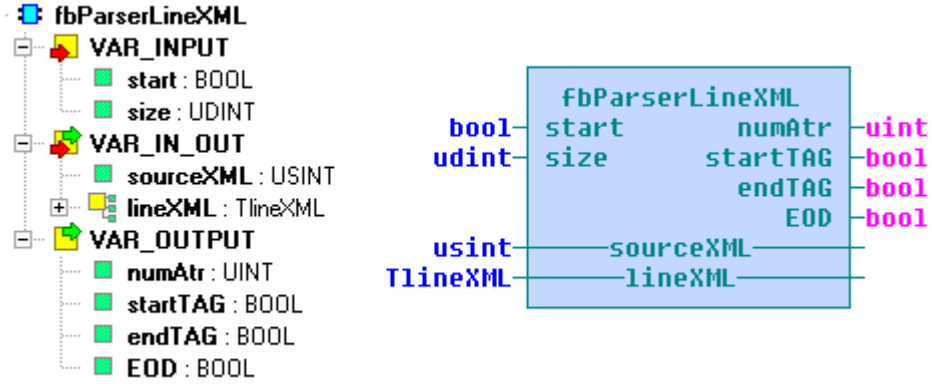
V knihovně XmlLib jsou definovány následující funkční bloky:



Funkční blok	*Popis*
fbParserLineXML	Zpracovává elementy XML dokumentu
fbComposerLineXML	Zapisuje elementy XML dokumentu

6.1 Funkční blok fbParserLineXML

Knihovna : *XmlLib*



Funkční blok *fbParserLineXML* slouží k rozebírání XML dokumentu po jednotlivých elementech. Na začátku rozebírání je nutné nastavit proměnnou *start* na hodnotu TRUE a proměnnou *size* na velikost proměnné, ve které je uložen XML dokument. Vlastní proměnná s XML dokumentem se předává na vstupu *sourceXML*. Výsledek zpracování se vrací ve struktuře *TlineXML* na vstupu *lineXML*. Následující volání s proměnnou *start* nastavenou na hodnotu FALSE vrací další elementy XML v pořadí, jak za sebou v dokumentu následují. Výstupy funkčního bloku dávají další informace o načteném elementu. Výstup *numAtr* udává počet atributů načteného tagu (0 až 10). Příznaky *startTAG* a *endTAG* v sobě nesou informaci o tom jestli se jedná o počáteční nebo koncoví tag. Pokud mají oba příznaky hodnotu TRUE jedná se o nepárový tag. Výstup *EOD* je nastaven v případě, že byly přečteny všechny elementy XML dokumentu.

Popis proměnných :

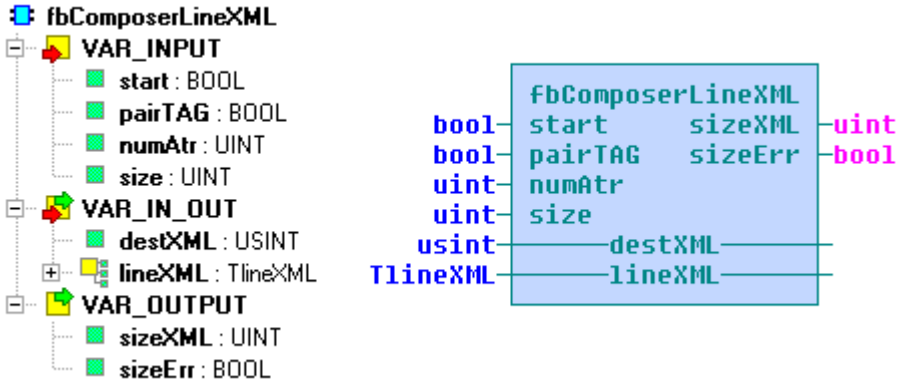
	Proměnná	*Typ*	*Význam*
VAR_INPUT			
	*start*	BOOL	TRUE začne procházet od začátku XML dat, FALSE pokračuje tam, kde se minule skončilo
	*size*	UDINT	Velikost XML dokumentu v bytech
VAR_IN_OUT			
	*sourceXML*	USINT	Proměnná obsahující XML dokument
	*lineXML*	TlineXML	Popis zpracovaného XML elementu
VAR_OUTPUT			
	*numAtr*	BOOL	Počet atributů aktuálně načteného tagu
	*startTAG*	BOOL	TRUE znamená začátek párového tagu (např. </ROOT>)
	*endTAG*	BOOL	TRUE znamená konec párového tagu (např. </ROOT>)
	*EOD*	BOOL	Konec dat, dokument byl přečten až do velikosti udané proměnnou *size*

Příklad použití (komentář „...“ je potřeba nahradit zpracováním XML tagů) :

```
VAR_GLOBAL
  xmlTemplate : ARRAY [0..3] OF STRING := ['<data>', '<date></date>',
     '<value valid=""></value>', '</data>'];
END_VAR

PROGRAM prgXmlExample
  VAR
    ParserLineXML : fbParserLineXML;
    lineXML : TlineXML;
  END_VAR
  VAR_TEMP
    i : UINT;
  END_VAR

  ParserLineXML(start := 1, size := SIZEOF(xmlTemplate),
    sourceXML := void(xmlTemplate),
    lineXML := lineXML);
  IF lineXML.tag = 'data' THEN
    WHILE NOT ParserLineXML.EOD AND lineXML.tag <> '/data' DO
      ParserLineXML(start := 0, sourceXML := void(xmlTemplate),
        lineXML := lineXML);
      (* ... *)
    END_WHILE;
  END_IF;

END_PROGRAM
```

6.2 Funkční blok fbComposerLineXML

Knihovna : *XmlLib*



Funkční blok *fbComposerLineXML* slouží k zápisu XML dokumentu do proměnné. Na začátku zápisu je nutné nastavit proměnnou *start* na hodnotu TRUE a proměnnou *size* na velikost proměnné, ve které je bude výsledný XML dokument. Cílová proměnná se předává na vstupu dest*XML*. Následující volání s proměnnou *start* nastavenou na hodnotu FALSE zapisuje další elementy XML za poslední zapsaný. Výsledný zápis je ovlivněn příznaky *pairTAG* a *numAtr*. Vstup *numAtr* udává počet atributů zapisovaného tagu (0 až 10). Příznaky *pairTAG* určuje zda bude zapsán tag jako párový (hodnota TRUE) nebo mu bude přidán na konec znak nepárového tagu (hodnota FALSE).

Popis proměnných :

	Proměnná	*Typ*	*Význam*
VAR_INPUT			
	start	BOOL	TRUE začne ukládat od začátku XML dat, FALSE pokračuje tam, kde se minule skončilo
	pairTAG	BOOL	Jedná se o párový tag
	numAtr	UINT	Počet atributů
	size	UINT	Délka zapisovaných dat (počet bytů)
VAR_IN_OUT			
	destXML	USINT	Proměnná, ve které jsou připravena zapisovaná data
	lineXML	TlineXML	Popis elementu, který bude zapsán do XML
VAR_OUTPUT			
	sizeXML	BOOL	Aktuální velikost XML
	sizeErr	BOOL	Nedostatek místa pro zápis elementu

Příklad programu s funkčním blokem *fbComposerLineXML:*

```
VAR_GLOBAL
  xmlResult : ARRAY [0..511] OF USINT;
END_VAR

PROGRAM prgXmlExample
  VAR
    ComposerLineXML : fbComposerLineXML;
    lineXML : TlineXML;
  END_VAR
  VAR_TEMP
    i : UINT;
  END_VAR

  FOR i := 0 TO 5 DO
    CASE i OF
      0 : lineXML.tag := 'data';
          ComposerLineXML.pairTAG := true;
          ComposerLineXML.numAtr := 0;
      1 : lineXML.tag := 'date';
      2 : lineXML.tag := '/date';
                  lineXML.txt  :=  DT_TO_STRINGF(GetDateTime(),   '%TYYYY/MM/DD-
      3 : lineXML.tag := 'value'; lineXML.atr[1] := 'valid';
          lineXML.val[1] := BOOL_TO_STRING(
            NOT(r0_p3_AI0.STAT.UNR OR r0_p3_AI0.STAT.OVR));
          ComposerLineXML.numAtr := 1;
          lineXML.txt := '';
      4 : lineXML.tag := '/value';
          lineXML.txt := REAL_TO_STRINGF(r0_p3_AI0.ENG, '%5.2f');
          ComposerLineXML.numAtr := 0;
      5 : lineXML.tag := '/data';
          lineXML.txt := '';
    END_CASE;
    ComposerLineXML(start := i = 0,
        size := SIZEOF(xmlResult)-1, destXML := void(xmlResult),
        lineXML := lineXML);
  END_FOR;

  xmlResult[ComposerLineXML.sizeXML] := 0;

END_PROGRAM
```

Výsledkem programu je následující XML dokument, který může vypadat například takto:

<data><date>2010/11/11-10:39:25</date><value valid=“1“>12.5</value></data>

7 PŘÍKLAD POUŽITÍ

Následující příklad ukazuje možnosti použití funkčních bloků *fbParserLineXML* a *fbComposerLineXML*. Pomocí bloku *fbParserLineXML* je vyčtena struktura XML dokumentu z proměnné *xmlTemplate*. Načtená struktura je doplňována programem a opět převáděna na XML dokument, blokem *fbComposerLineXML*.

Program nejprve zkontroluje, zda-li je počáteční tag <data>. Pokud ano, hledá koncoví tag </date> před, který přidá datum zformátovaný funkcí z knihovny ToStringLib. Když narazí na tag <value> zkusí vyhledat atribut valid, do kterého doplní stav z analogového vstupu. Při nalezení koncového tagu </value> je před něj doplněn text z hodnotou analogového vstupu.

Po sestavení celého dokumentu je přidán na konec znak 0.

```
VAR_GLOBAL
  xmlTemplate : ARRAY [0..3] OF STRING :=
    ['<data>',
     '<date></date>',
     '<value valid=""></value>',
     '</data>'];
  xmlResult : ARRAY [0..511] OF USINT;
END_VAR

PROGRAM prgXmlExample
  VAR
    ParserLineXML : fbParserLineXML;
    ComposerLineXML : fbComposerLineXML;
    lineXML : TlineXML;
  END_VAR
  VAR_TEMP
    i : UINT;
  END_VAR

  ParserLineXML(start := 1, size := SIZEOF(xmlTemplate),
    sourceXML := void(xmlTemplate),
    lineXML := lineXML);
  IF lineXML.tag = 'data' THEN
    ComposerLineXML(start := 1,
      pairTAG := NOT (ParserLineXML.startTAG AND ParserLineXML.endTAG),
      numAtr := 0, size := SIZEOF(xmlResult)-1, destXML := void(xmlResult),
      lineXML := lineXML);
    WHILE NOT ParserLineXML.EOD AND lineXML.tag <> '/data' DO
      ParserLineXML(start := 0, sourceXML := void(xmlTemplate),
        lineXML := lineXML);
      IF lineXML.tag = '/date' THEN
        lineXML.txt := DT_TO_STRINGF(GetDateTime(), '%TYYYY/MM/DD-hh:mm:ss');
      END_IF;
      IF  lineXML.tag = 'value' THEN
        FOR i := 1 TO ParserLineXML.numAtr DO
          IF lineXML.atr[i] = 'valid' THEN
            lineXML.val[i] := BOOL_TO_STRING(
                              NOT(r0_p3_AI0.STAT.UNR OR r0_p3_AI0.STAT.OVR));
            EXIT;
          END_IF;
        END_FOR;
      END_IF;
      IF  lineXML.tag = '/value' THEN
        lineXML.txt := REAL_TO_STRINGF(r0_p3_AI0.ENG, '%5.2f');
      END_IF;
      ComposerLineXML(start := 0,
        pairTAG := NOT (ParserLineXML.startTAG AND ParserLineXML.endTAG),
        numAtr := ParserLineXML.numAtr, destXML := void(xmlResult),
        lineXML := lineXML);
    END_WHILE;
  END_IF;

  xmlResult[ComposerLineXML.sizeXML] := 0;

END_PROGRAM
```

Objednávky a informace:
Teco a. s. Havlíčkova 260, 280 58 Kolín 4, tel. 321 737 611, fax 321 737 633

TXV 003 63.01