

Natural Simple XML Parser

The following topics are covered:

- Parser Description and Example
 - Parser Restrictions
-

Parser Description and Example

The Natural simple XML Parser enables you to parse XML documents with standard Natural programs. The parser sends an event, or runs an internal subroutine callback if the next part of the document is parsed. The inline subroutine "CALLBACK" is called with the name of the current element, text, comment within an xpath-like syntax. The parser engine is included as copy code "PARSER_X". If an error occurs during parsing, e.g. the document is not wellformed, the "PARSER_ERROR" inline subroutine is called and then the parser is canceled with "ESCAPE SUBROUTINE" (see also Parser Restrictions).

For extended error handling, it is possible to change the operand6 "Error Message Text" and operand7 "Error Number" to a value less than or equal to -9000. Then the "PARSER_ERROR" inline subroutine is called and the (sub)program is canceled with "ESCAPE SUBROUTINE". If other values are less than or equal to -8000, only the parser is canceled with "ESCAPE SUBROUTINE".

The major variables of the parser are defined at the Local Data Area "PARSER-X".

The parser copycode takes the following operands:

Operand	Format/Length	Description
1	A	XML file to be parsed
2	A	ex-XPATH to represent element structure
3	A1	Type of the XPATH content: ? Processing instruction D DOCTYPE ! Comment C CDATA section T Starting Tag @ Attribute / Close Tag
4	A	Parsed Data
5	L	Is TRUE if Parsed Data is empty
6	A	Error Message Text
7	I4	Error Number

Return value of the XPATH data:

ex-XPath	XML Structure
?	<? ... ?>
!DOCTYPE	<!DOCTYPE ... >
!DOCTYPE[<!DOCTYPE .. [...]>
![CDATA[<![CDATA[...]]>
!--	<!-- -->
!	<! .. >
doc	<doc>
doc doc/foo doc/foo/\$ doc/foo// doc//	<doc><foo>text</foo></doc>
doc doc/@a1 doc//	<doc a1="a" />
doc doc/@a1 doc/@a2 doc/\$ doc//	<doc a1="a" a2="b">text</doc>
doc doc/\$ doc/foo doc/foo/\$ doc/foo// doc/\$ doc//	<doc> <foo>text</foo> </doc>
doc doc/![CDATA[doc//	<doc><![CDATA[...]]></doc>
doc doc/!-- doc//	<doc><!-- ... --></doc>

Program Example:

```

* -----
* CLASS NATURAL XML TOOLKIT - UTILITIES
*
*     PARSE
*
* DESCRIPTION
*     Parse given XML
*
*

```

```

* AUTHOR          SAG    01.2006
*
* VERSION         6.2.
*
* (c) Copyright Software AG 2006. All rights reserved.
*
* -----
*
DEFINE DATA LOCAL
1 XML_PARSER_INPUT          (A) DYNAMIC
1 XML_PARSER_ERROR_TEXT    (A253)
1 XML_PARSER_RESPONSE      (I4)
LOCAL USING PARSE-X        /* parser internal data - do not change
LOCAL
1 XML_PARSER_XPATH         (A) DYNAMIC
1 XML_PARSER_XPATH_TYPE    (A1)
1 XML_PARSER_CONTENT       (A) DYNAMIC
1 XML_PARSER_CONTENT_IS_EMPTY (L)
*
1 ANFANG                   (T)
* OUT                       (A) DYNAMIC
1 OUT                      (A126)
*
END-DEFINE
*
FORMAT (0) LS=128 PS=40
*
DEFINE WORK FILE 12 "E:\EMPLOYEE1.XML" TYPE "UNFORMATTED"
READ WORK FILE 12 XML_PARSER_INPUT
END-WORK
CLOSE WORK FILE 12
*
*
* ----- INCLUDE THE PARSE
INCLUDE PARSE_X 'XML_PARSER_INPUT' /* XML file to be parsed
'XML_PARSER_XPATH' /* XPATH to represent element...
'XML_PARSER_XPATH_TYPE' /* Type of callback
'XML_PARSER_CONTENT' /* Content of element found
'XML_PARSER_CONTENT_IS_EMPTY' /* Is TRUE if element is empty
'XML_PARSER_ERROR_TEXT' /* error Message
'XML_PARSER_RESPONSE' /* Error NR; 0 = OK
*
*
DEFINE SUBROUTINE CALLBACK
IF XML_PARSER_CONTENT_IS_EMPTY THEN
  IF XML_PARSER_XPATH_TYPE NE "T" AND XML_PARSER_XPATH_TYPE NE "/" THEN
    COMPRESS XML_PARSER_XPATH "(NULL)" INTO OUT WITH DELIMITER "="
  ELSE
    OUT := XML_PARSER_XPATH
  END-IF
ELSE
  COMPRESS XML_PARSER_XPATH XML_PARSER_CONTENT INTO OUT WITH DELIMITER "="
END-IF
WRITE OUT
END-SUBROUTINE
/*
DEFINE SUBROUTINE PARSE_ERROR
OUT := XML_PARSER_ERROR_TEXT
WRITE OUT
END-SUBROUTINE
END

```

With a given result document from Tamino for the Employee data, the result of this program looks like this:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<Employee xmlns:ino="http://namespaces.softwareag.com/tamino/response2" ino:id="560"
Personnel-ID="20006900">
  <Full-Name>
    <First-Name>JOE</First-Name>
    <Name>ATHERTON</Name>
  </Full-Name>
  <Mar-Stat>S</Mar-Stat>
  <Sex>M</Sex>
  <Birth>1941-02-21</Birth>
  <Full-Address>
    <Address-Line>11603 HUNTERS GREEN</Address-Line>
    <Address-Line>SYRACUSE</Address-Line>
    <Address-Line>NY</Address-Line>
    <City>SYRACUSE</City>
    <Zip>13201</Zip>
    <Post-Code>13201</Post-Code>
    <Country>USA</Country>
  </Full-Address>
  <Telephone>
    <Phone>173-9859</Phone>
    <Area-Code>315</Area-Code>
  </Telephone>
  <Dept>TECH10</Dept>
  <Job-Title>ANALYST</Job-Title>
  <Income>
    <Curr-Code>USD</Curr-Code>
    <Salary>43000</Salary>
  </Income>
  <Income>
    <Curr-Code>USD</Curr-Code>
    <Salary>39500</Salary>
  </Income>
  <Income>
    <Curr-Code>USD</Curr-Code>
    <Salary>36700</Salary>
  </Income>
  <Income>
    <Curr-Code>USD</Curr-Code>
    <Salary>34400</Salary>
  </Income>
  <Income>
    <Curr-Code>USD</Curr-Code>
    <Salary>32600</Salary>
  </Income>
  <Leave-Data>
    <Leave-Due>19</Leave-Due>
    <Leave-Taken>4</Leave-Taken>
  </Leave-Data>
  <Leave-Booked>
    <Leave-Start>19980112</Leave-Start>
    <Leave-End>19980112</Leave-End>
  </Leave-Booked>
  <Leave-Booked>
    <Leave-Start>19980605</Leave-Start>
    <Leave-End>19980605</Leave-End>
  </Leave-Booked>
  <Leave-Booked>
    <Leave-Start>19980916</Leave-Start>
```

```
<Leave-End>19980916</Leave-End>
</Leave-Booked>
<Lang>ENG</Lang>
</Employee>
```

Note:

There is no line break in the whole document.

The result of the above Natural program looks like this:

```
?=xml version="1.0" encoding="ISO-8859-1"
Employee
Employee/@xmlns:ino=http://namespaces.softwareag.com/tamino/response2
Employee/@ino:id=560
Employee/@Personnel-ID=20006900
Employee/Full-Name
Employee/Full-Name/First-Name
Employee/Full-Name/First-Name/$=JOE
Employee/Full-Name/First-Name//
Employee/Full-Name/Name
Employee/Full-Name/Name/$=ATHERTON
Employee/Full-Name/Name//
Employee/Full-Name//
Employee/Mar-Stat
Employee/Mar-Stat/$=S
Employee/Mar-Stat//
Employee/Sex
Employee/Sex/$=M
Employee/Sex//
Employee/Birth
Employee/Birth/$=1941-02-21
Employee/Birth//
Employee/Full-Address
Employee/Full-Address/Address-Line
Employee/Full-Address/Address-Line/$=11603 HUNTERS GREEN
Employee/Full-Address/Address-Line//
Employee/Full-Address/Address-Line
Employee/Full-Address/Address-Line/$=SYRACUSE
Employee/Full-Address/Address-Line//
Employee/Full-Address/Address-Line
Employee/Full-Address/Address-Line/$=NY
Employee/Full-Address/Address-Line//
Employee/Full-Address/City
Employee/Full-Address/City/$=SYRACUSE
Employee/Full-Address/City//
Employee/Full-Address/Zip
Employee/Full-Address/Zip/$=13201
Employee/Full-Address/Zip//
Employee/Full-Address/Post-Code
Employee/Full-Address/Post-Code/$=13201
Employee/Full-Address/Post-Code//
Employee/Full-Address/Country
Employee/Full-Address/Country/$=USA
Employee/Full-Address/Country//
Employee/Full-Address//
Employee/Telephone
Employee/Telephone/Phone
Employee/Telephone/Phone/$=173-9859
Employee/Telephone/Phone//
Employee/Telephone/Area-Code
Employee/Telephone/Area-Code/$=315
Employee/Telephone/Area-Code//
```

```
Employee/Telephone//
Employee/Dept
Employee/Dept/=$=TECH10
Employee/Dept//
Employee/Job-Title
Employee/Job-Title/=$=ANALYST
Employee/Job-Title//
Employee/Income
Employee/Income/Curr-Code
Employee/Income/Curr-Code/=$=USD
Employee/Income/Curr-Code//
Employee/Income/Salary
Employee/Income/Salary/=$=43000
Employee/Income/Salary//
Employee/Income//
Employee/Income
Employee/Income/Curr-Code
Employee/Income/Curr-Code/=$=USD
Employee/Income/Curr-Code//
Employee/Income/Salary
Employee/Income/Salary/=$=39500
Employee/Income/Salary//
Employee/Income//
Employee/Income
Employee/Income/Curr-Code
Employee/Income/Curr-Code/=$=USD
Employee/Income/Curr-Code//
Employee/Income/Salary
Employee/Income/Salary/=$=36700
Employee/Income/Salary//
Employee/Income//
Employee/Income
Employee/Income/Curr-Code
Employee/Income/Curr-Code/=$=USD
Employee/Income/Curr-Code//
Employee/Income/Salary
Employee/Income/Salary/=$=34400
Employee/Income/Salary//
Employee/Income//
Employee/Income
Employee/Income/Curr-Code
Employee/Income/Curr-Code/=$=USD
Employee/Income/Curr-Code//
Employee/Income/Salary
Employee/Income/Salary/=$=32600
Employee/Income/Salary//
Employee/Income//
Employee/Leave-Data
Employee/Leave-Data/Leave-Due
Employee/Leave-Data/Leave-Due/=$=19
Employee/Leave-Data/Leave-Due//
Employee/Leave-Data/Leave-Taken
Employee/Leave-Data/Leave-Taken/=$=4
Employee/Leave-Data/Leave-Taken//
Employee/Leave-Data//
Employee/Leave-Booked
Employee/Leave-Booked/Leave-Start
Employee/Leave-Booked/Leave-Start/=$=19980112
Employee/Leave-Booked/Leave-Start//
Employee/Leave-Booked/Leave-End
Employee/Leave-Booked/Leave-End/=$=19980112
Employee/Leave-Booked/Leave-End//
```

```

Employee/Leave-Booked//
Employee/Leave-Booked
Employee/Leave-Booked/Leave-Start
Employee/Leave-Booked/Leave-Start/$=19980605
Employee/Leave-Booked/Leave-Start//
Employee/Leave-Booked/Leave-End
Employee/Leave-Booked/Leave-End/$=19980605
Employee/Leave-Booked/Leave-End//
Employee/Leave-Booked//
Employee/Leave-Booked
Employee/Leave-Booked/Leave-Start
Employee/Leave-Booked/Leave-Start/$=19980916
Employee/Leave-Booked/Leave-Start//
Employee/Leave-Booked/Leave-End
Employee/Leave-Booked/Leave-End/$=19980916
Employee/Leave-Booked/Leave-End//
Employee/Leave-Booked//
Employee/Lang
Employee/Lang/$=ENG
Employee/Lang//
Employee//

```

Parser Restrictions

The parser does not handle:

- Composition of a tag (incl. processing instruction). Only start-tag must be equal to end-tag (incl. processing instruction).

Example:

```

<.doc></.doc> <!-- invalid character in tag -->
<doc><? ?></doc> <!-- invalid whitespace -->
<doc>&#RE;</doc> <!-- invalid character in tag -->

```

- Character or entity references

Example:

```

<doc>& no refc</doc> <!-- missing semicolon --> <doc a1=v1></doc>
<!-- string literal expected -->

```

- Exact handling of CDATA-Sections

Example:

```

<doc><![CDATA [ stuff]]></doc> <!-- must be CDATA[ -->

```

- Content of an entity/processing instruction

Example:

```

<doc>]]></doc> <!-- ]] not allowed -->

```

- Number of tags/attributes

- Headerinformation
- Unicode-charset (supports ISO-8859-1)