

# Using the XML Toolkit

The following topics are covered:

- Prerequisites
  - Work File Processing
  - Print Files
  - Invoking the Application
  - PF-Key Assignments
- 

## Prerequisites

### Storage Requirements

Depending on the size, complexity or the recursion depth of the processed XML DTD, the XML Toolkit might require up to several hundred kilobytes of space in the DATSIZE buffer.

Parsing an XML document using the generated callback routine requires that the entire document is contained in a dynamic variable.

### Schema Support

Schemas are not supported by the XML Toolkit on mainframe computers; only DTDs can be processed.

### Codepage Support

For the use of the XML Toolkit, the Natural session must be driven with code page support enabled. See Natural profile parameter CP.

## Work File Processing

By default, the XML Toolkit uses Natural Work File 12 as input or output file for DTDs and Work File 13 as error logfile.

The sample parser program, delivered with the XML Toolkit, reads the XML document to be parsed from Work File 12 and writes the output of the parse process to Work File 13.

### Work File Support

Running the XML Toolkit under TP monitors which do not support Natural work files is possible using PC work files, if you are working with a PC on which Entire Connection is in use.

XML Documents which shall be parsed using the generated callback routine or which shall be generated using the generated serialization code can be accessed via work files as well. Data must be accessed with a variable of data type ALPHA DYNAMIC. The work file should be of data type UNFORMATTED.

PC files can be accessed using the subprogram XML2PCWR provided in the SYSEXXT library. XML2PCWR writes or reads an ALPHA DYNAMIC variable to or from a PC work file.

### Calling Conventions for XML2PCWR:

```
CALLNAT 'XML2PCWR' XML-PAGE FILENUMBER OPERATION RETCODE
```

### Parameter Definition:

```
DEFINE DATA LOCAL
  1 XML-PAGE          (A)      DYNAMIC          /* XML page
  1 FILENUMBER        (I2)
  1 OPERATION         (A2)
  1 RETCODE           (I4)          /* must be 0!
END-DEFINE
```

### Example:

The following example writes the generated document to Standard Work File 10 and to PC Work File 15:

```
* -----
* CLASS NATURAL XML TOOLKIT - UTILITIES
*
* SDEMO_P1
*
* DESCRIPTION
* Serialize a given Data structure.
*
*
* AUTHOR SAG 01.2005
*
* VERSION 4.12.
*
* (c) Copyright Software AG 2001-2005. All rights reserved.
*
* -----
*
DEFINE DATA
LOCAL USING EMPL /* add generated data structure
LOCAL
1 XML (A) DYNAMIC
*
1 OUT (A72)
1 II (I4)
*
1 #CX (I4)
1 #CY (I4)
1 #CZ (I4)
1 FILENUMBER (I2)
1 OPERATION (A2)
1 RETCODE (I4)
END-DEFINE
/*[ initialize
EMPLOYEE.PERSONNEL-ID := 4711
*
EMPLOYEE.FIRST-NAME := "ADKINSON"
EMPLOYEE.NAME := "MARTHA"
*
EMPLOYEE.CÂ$ADDRESS-LINE := 2
EMPLOYEE.ADDRESS-LINE(1) := "8603 GARLAND COURT"
EMPLOYEE.ADDRESS-LINE(2) := "FRAMINGHAM"
```

```

EMPLOYEE.ADDRESS-LINE(2) := "MA"
EMPLOYEE.CITY := "FRAMINGHAM"
EMPLOYEE.ZIP := "17010"
EMPLOYEE.COUNTRY := "USA"
*
EMPLOYEE.AREA-CODE := "617"
EMPLOYEE.PHONE := "210-4703"
*
EMPLOYEE.JOB-TITLE := "MANAGER"
EMPLOYEE.CÂ$INCOME := 2
EMPLOYEE.SALARY(1) := 47000
EMPLOYEE.CÂ$BONUS(1) := 2
EMPLOYEE.BONUS(1,1) := 10500
EMPLOYEE.BONUS(1,2) := 7875
*
EMPLOYEE.SALARY(2) := 47000
EMPLOYEE.CÂ$BONUS(2) := 1
EMPLOYEE.BONUS(2,1) := 35700
*
INCLUDE EMPL-C "XML" "#CX" "#CY" "#CZ"
/* add generated Serialize
/*]
*
ASSIGN FILENUMBER = 15 /* PC FILE
ASSIGN OPERATION = 'W'
CALLNAT 'XML2PCWR' XML FILENUMBER OPERATION RETCODE
PRINT XML
DEFINE WORK FILE 10 TYPE 'UNFORMATTED' /* STD WORK FILE
WRITE WORK FILE 10 VARIABLE XML
CLOSE WORK FILE 10
*
END

```

## Print Files

The XML Toolkit writes to Report 2.

## Invoking the Application

The XML toolkit is included in the library SYSEXXT.

### To use the XML Toolkit

- In the Natural command line, enter LOGON SYSEXXT.

Enter Menu.

The Main Menu is displayed.

```

10:22:54          *** NATURAL XML Toolkit ***          2007-01-19
                  - Main Menu -                      Library SYSEXXT

Code  Function

      L  Generate from Natural Data Structure
      D  Generate from Document Type Definition
      O  Set up Specific Generation Options

Function Code .. _

Press PF4 or PF5 to start generation.
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      Exit  LDA  DTD                      Optio      Canc

```

The following functions are available:

- Generate from Natural Data Structure Uses the Natural Data Area as a data source.
- Generate from Document Type Definition Uses the Document Type Definition (.dtd) as a data source.
- Set up Specific Generation Options

For details, refer to the corresponding sections.

## PF-Key Assignments

The following function keys are used for navigation and processing.

<b>PF1</b>	Help	Context-related help. For more information, refer to the online documentation.
<b>PF3</b>	Exit	On an options map: closes the function and saves the changes. On a generation map: closes the function after a generation is done. On the Main Menu map: closes the application.
<b>PF7</b>	Prev	Previous step (previous map).
<b>PF8</b>	Next	Next step (next map).
<b>PF9</b>	Finis(h)	Closes the function after a generation is done.
<b>PF12</b>	Cancel	Closes the function without saving the changes. Closes the function if no generation has already be done.