Natural XML Toolkit

Description

Natural 5 XML Toolkit goal is to provide additional XML functionality with Natural and improve the integration of Natural applications with XML without using external software components like "msxml". It shall be understood as an intermediate step before implementing full XML functionality in Natural's language.

GENERAL ARCHITECTURE

The Natural XML Toolkit will be provided with Natural 5 for Windows and consists of a collection of Natural programs some of these will be delivered in source. The Toolkit programs may be integrated into customer applications to provide access to XML data or to deliver data from Natural in XML format. Together with the Toolkit programs Natural 5 includes example programs and Natural dialogs to control the Toolkit. The XML Toolkit and the new statement REQUEST DOCUMENT provides access to any source in the Internet as sole basis for the implementation of applications that uses XML in Natural.

THE NATURAL XML Toolkit

The Natural XML Toolkit is implemented as a Natural Dialog that calls the functions listed below:

XML Toolkit functions

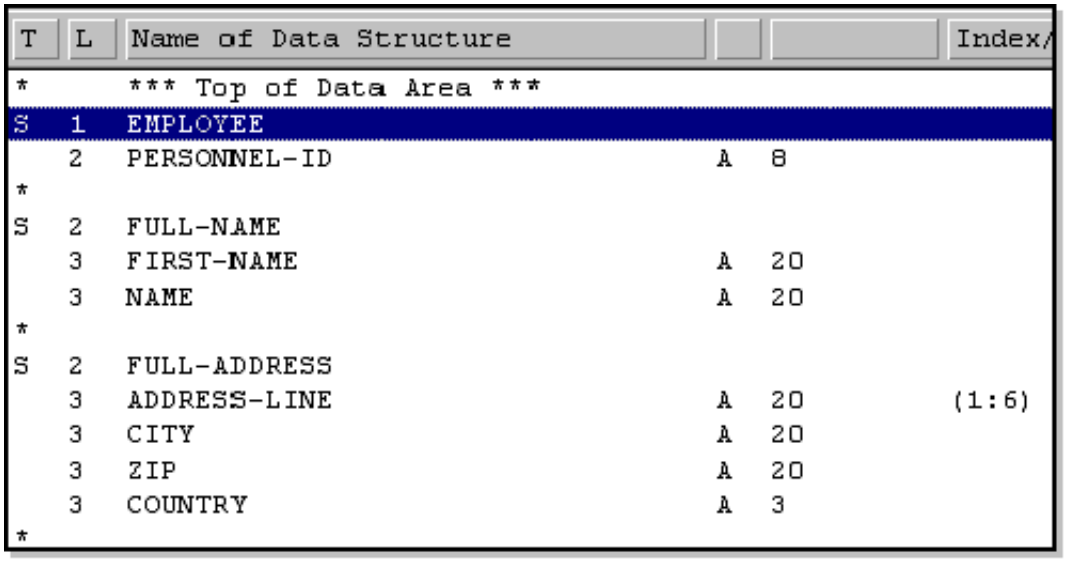
1. Mapping of Natural Data Definition to DTD and vice versa.
   Document type descriptions are most commonly used to describe the structure of a XML document.
2. XML Token -> NAT Data
   After creating the Natural data structure the XML document has to be parsed and saved into the data structure. An implementation for the delivered XML parser callback routine will be generated. This callback assign the value of a data element to the corresponding data structure.
3. NAT Data -> XML Document ("Serialize")
   Serialization is the process to take the data stored in the Natural data structures and create a XML document according to the description in the DTD.

A Natural dialog implements the user interface to the XML Toolkit functions. The DTD will be accessed as a work file and the generated Natural objects will be saved directly to the Natural system file.

MAP NATURAL DATA DEFINITIONS TO DTD

This mapping is the first step to bind Natural data structures to XML tags and is required to implement a representation of Natural data as XML tags. The example below shows the mapping as well as some obvious differences between Natural and a DTD.

Natural PDA



T	L	Name of Data Structure			Index/
*		*** Top of Data Area ***			
S	1	EMPLOYEE			
	2	PERSONNEL-ID	A	8	
*					
S	2	FULL-NAME			
	3	FIRST-NAME	A	20	
	3	NAME	A	20	
*					
S	2	FULL-ADDRESS			
	3	ADDRESS-LINE	A	20	(1:6)
	3	CITY	A	20	
	3	ZIP	A	20	
	3	COUNTRY	A	3	
*					

Generated DTD

```
<!ELEMENT EMPLOYEE (PERSONNEL-ID, FULL-NAME, FULL-ADDRESS, TELEPHONE, INCOME* )>

<!ELEMENT PERSONNEL-ID (#PCDATA ) >

<!ELEMENT FULL-NAME (FIRST-NAME, NAME )>
         <!ELEMENT FIRST-NAME  (#PCDATA )>
         <!ELEMENT NAME  (#PCDATA )>

<!ELEMENT FULL-ADDRESS (ADDRESS-LINE*, CITY, ZIP, COUNTRY )>
         <!ELEMENT ADDRESS-LINE  (#PCDATA )>
         <!ELEMENT CITY  (#PCDATA )>
         <!ELEMENT ZIP  (#PCDATA )>
         <!ELEMENT COUNTRY  (#PCDATA )>
...
```

The generated DTD will be used later on during serialization to a XML document. (see below).

SERIALIZE THE DATA TO XML

During execution of a Natural program the content of the data defined in the DEFINE DATA statement will be filled with "real" content. This content will be written to a dynamic variable in XML format during serialization and will be used the formerly generated DTD as input.

The XML Toolkit generates the program to serialize the data.

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<EMPLOYEE PERSONNEL-ID="30016509">
<FULL-NAME>
  <FIRST-NAME>ELSPETH</FIRST-NAME>
  <NAME>TROWBRIDGE</NAME>
</FULL-NAME>
<FULL-ADDRESS>
  <ADDRESS-LINE>91 BACK LANE</ADDRESS-LINE>
  <ADDRESS-LINE>BILSTON</ADDRESS-LINE>
  <ADDRESS-LINE>STAFFORDSHIRE</ADDRESS-LINE>
  <CITY>BILSTON</CITY>
  <ZIP>ST2 3KA</ZIP>
  <COUNTRY>UK</COUNTRY>
</FULL-ADDRESS>
<TELEPHONE>
```



```
  <PHONE>863322</PHONE>
  <AREA-CODE>0602</AREA-CODE>
</TELEPHONE>
...
```

MAP DTD TO NATURAL DATA DEFINITIONS

The mapping of a DTD to Natural data structures again shows differences. The DTD does not specify how many person records will be included in the XML document, therefore the Toolkit assumes that a maximum number of "v" persons will be included in the XML document. The application programmer might know the exact number and the data structure can be adapted accordingly. A similar limitation exists with the length of the data. The DTD does not include information about the length of the data in person record. Therefore the Toolkit creates fields in the data structure with a length of 253, the current maximum.

```
* DTD E:\SAG\nat\5.1.1\fnat\SYSEXXTG\RES\empl.dtd
COMPRESS &1& '<EMPLOYEE'
  ' PERSONNEL-ID="'EMPLOYEE.PERSONNEL-ID '"'
  '>' INTO &1& LEAVING NO
/* now the children
COMPRESS &1& '<FULL-NAME'
  '>' INTO &1& LEAVING NO
/* now the children
COMPRESS &1& '<FIRST-NAME'
  '>'
  EMPLOYEE.FIRST-NAME
  '</FIRST-NAME>' INTO &1& LEAVING NO
COMPRESS &1& '<NAME'
  '>'
  EMPLOYEE.NAME
  '</NAME>' INTO &1& LEAVING NO
/*
COMPRESS &1& '</FULL-NAME>' INTO &1& LEAVING NO
COMPRESS &1& '<FULL-ADDRESS'
  '>' INTO &1& LEAVING NO
/* now the children
FOR &2& = 1 TO EMPLOYEE.C@ADDRESS-LINE
  COMPRESS &1& '<ADDRESS-LINE'
    '>'
    EMPLOYEE.ADDRESS-LINE(&2&)
    '</ADDRESS-LINE>' INTO &1& LEAVING NO
END-FOR
...
```

PARSE XML FILE AND ASSIGN TO NATURAL DATA

The XML Toolkit generates Natural code based on the DTD. This code represents a subroutine to be called from the parser to assign the content of the tags in the XML document to the corresponding data structure.

```
* DTD E:\SAG\nat\5.1.1\fnat\SYSEXXTG\RES\empl.dtd
DECIDE ON FIRST &1&
  VALUE 'EMPLOYEE'
    RESET INITIAL EMPLOYEE
  VALUE 'EMPLOYEE/@PERSONNEL-ID'
    /* #REQUIRED
    EMPLOYEE.PERSONNEL-ID := &3&
  VALUE 'EMPLOYEE/FULL-NAME'
    IGNORE
  VALUE 'EMPLOYEE/FULL-NAME/FIRST-NAME'
    IGNORE
  VALUE 'EMPLOYEE/FULL-NAME/FIRST-NAME/$'
    EMPLOYEE.FIRST-NAME := &3&
  VALUE 'EMPLOYEE/FULL-NAME/NAME'
```



```
    IGNORE
  VALUE 'EMPLOYEE/FULL-NAME/NAME/$'
    EMPLOYEE.NAME := &3&

...
```

OUTLOOK

The XML Toolkit is another step forward to full XML support with Natural. The XML Toolkit might be extended after the first release. Programs to map Natural data to a Tamino Schema are subjects of investigation. However the main target is to implement XML functionality in one of the next releases as part of Natural's powerful language.

Features

- Natural-based XML parser using dynamic variables.

Features Wizzard

- Conversion of Natural data structures into DTD definitions.
- Generation of Compress statements to save a Natural data structure as an XML document
- Generation of callback for the Natural-based parser.

Prerequisite

- Natural 5.1

Running the Application

Run dialog `XML2NAT`.

Use Natural Data Area as Data Source

1. Start XML2NAT Dialog
2. Start wizard at **Generate > From Natural Data Structure**

Use Data Type Definition (.dtd) as Data Source

1. Start XML2NAT Dialog
2. Start wizard at **Generate > From Data Type Definiton**

Help

From all dialogs, help can be called by choosing **F1**.
All files needed to generate the HTML-based help used can be found at:
`..\SYSEXXT\RES\HELP`

For a new compilation, Microsoft HTML Help Workshop 1.2 is needed.

Note:
The warning *HHC3015: Warning: An alias ...* during compilation can be ignored.