Tips

Table of contents

1. Tips

1.1. 1. Emacs + nxml-mode

Emacs + nxml-mode ####### **mapping definition** # validation ################ nxml-mode ##### rnc ################# $TABLETOOLS_HOME/schema/rnc ##########

```
$ ant rnc
```

mapping definition # ##############msCodeGenerator #################### ############## datatypeLibrary # **URI** # type ########### ##################################

1.2. 2. ########### value

#####################

```
... <txt:value>
</txt:value>
```

##########

```
<txt:value>\n</txt:value>
```

1.3. 3.

##

```
$ createsample -lang -type ums_file output_directry
```

###

```
  + Sample

    + clng
      - SampleMain.c
      - SampleTest.c
      - Makefile
      - schemas.xml
      - Sample.ums

    + java
      - SampleMain.java
      - SampleTest.java
      - build.xml
```

```
        - schemas.xml
        - Sample.ums

      + data
```

################

1.4. 4. validation

####### #### ### validation ################ ####### ':' ############################# JAVA ##############

```

<?xml version="1.0" encoding="utf-8"?>
<grammar xmlns="http://ums.isas.jaxa.jp/0.4"
         datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes">
  <start>
import jp.jaxa.isas.ums.runtime.*;
import jp.jaxa.isas.ums.m3.*;

class ValidateSample {

  static void validate( byte[] ums__buffer, int ums__bitlen ) throws
UMSException {
    <defineMapping direction="decode">
      <dat:byte encode="txt" xmlns:dat="http://ums.isas.jaxa.jp/0.4/dat">
        <dat:list separator=":">
          <dat:data type="string"/>
          <dat:data type="string"/>
          <dat:data type="string"/>
          <dat:data type="string"/>
          <dat:data type="string"/>
          <dat:data type="string"/>
          <dat:data type="string"/>
          <dat:data type="string"/>
          <dat:data type="string"/>
        </dat:list>
      </dat:byte>
    </defineMapping>
  }

  /* ############ */
  public static void main(String[] args) {

    UMSLibrary.tableTools_init();

    String fileName = args[0];
    String inputBuffer;
    byte[] buffer = null;
    int inBitlen = 0;

    try {
      java.io.BufferedReader br
        = new java.io.BufferedReader( new java.io.FileReader( fileName ) );

      while( ( inputBuffer = br.readLine() ) != null ) {

        inBitlen = inputBuffer.length() * 8;

        try {
          buffer = inputBuffer.getBytes( "US-ASCII" );
          validate( buffer, inBitlen );
        } catch ( UMSException ex ) {
          ex.print( buffer, inBitlen );
          ex.printStackTrace( System.err );
        }
      }
      br.close();

    } catch ( java.io.IOException ex ) {
      ex.printStackTrace( System.err );
    }
    UMSLibrary.tableTools_end();
  }
}
  </start>
</grammar>
```

###C###############

```

<?xml version="1.0" encoding="UTF-8"?>
<grammar xmlns="http://ums.isas.jaxa.jp/0.4"
         datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes">
  <start>

void validate( char *ums__buffer, int ums__bitlen, ums__exception_t
*ums__ex ) {
    <defineMapping direction="decode">
      <dat:byte encode="txt" xmlns:dat="http://ums.isas.jaxa.jp/0.4/dat">
        <dat:list separator=":">
          <dat:data type="string"/>
          <dat:data type="string"/>
          <dat:data type="string"/>
          <dat:data type="string"/>
          <dat:data type="string"/>
          <dat:data type="string"/>
          <dat:data type="string"/>
          <dat:data type="string"/>
          <dat:data type="string"/>
        </dat:list>
      </dat:byte>
    </defineMapping>
}

/* ############ */
int main() {
  char in_buffer[BUFFER_SIZE];
  int  in_bitlen;
  ums__exception_t ums__ex;

  tableTools_init();

  while ( fgets( in_buffer, BUFFER_SIZE, stdin ) != NULL ) {

    in_bitlen = strlen( in_buffer ) * 8 - 8;

    in_buffer[in_bitlen/8] = 0;
    printf( "input(%d*8+%d):<%s>\n", in_bitlen / 8, in_bitlen % 8 ,
in_buffer );

    initException( &ums__ex );

    validate(input_buffer, in_bitlen, &ex);

    if ( ums__ex.occured != UMS__STATE_OK ) {
      ums__exception_print( &ums__ex, in_buffer, in_bitlen );
      continue;
    }
  }

  tableTools_end();

  return 0;
}
  </start>
</grammar>
```

1.5. 5. enocde # decode

'define'## # ## **'ref'##** # ######################## ##############################

```
<?xml version="1.0" encoding="UTF-8"?>
<grammar xmlns="http://ums.isas.jaxa.jp/0.4"
         datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes">
  <start>

    <java:class name="Sample"
xmlns:java="http://ums.isas.jaxa.jp/0.4/java">

      <defineVariables>
        <java:var class="String" name="sData"/>
        <java:var type="int"     name="iData"/>
        <java:var type="double"  name="dData"/>
      </defineVariables>

      <defineFunctions>
        <java:function name="decode">
          <java:arg       type="byte[]" name="ums__buffer" direction="in"/>
          <java:arg       type="int"    name="ums__bitlen" direction="in"/>
          <java:return    type="void"/>
          <java:exception type="UMSException"/>

          <defineMapping direction="decode">

            <ref name="mapping">

          </defineMapping>
        </java:function>

        <java:function name="encode">
          <java:arg       type="byte[]" name="ums__buffer"
direction="out"/>
          <java:arg       type="int[]"  name="ums__bitlen"
direction="inout"/>
          <java:return    type="void"/>
          <java:exception type="UMSException"/>

          <defineMapping direction="encode">

            <ref name="mapping">

          </defineMapping>
        </java:function>

      </defineFunctions>
    </java:class>
  </start>

  <define name="mapping">
    <dat:byte encode="txt" xmlns:dat="http://ums.isas.jaxa.jp/0.4/dat">
      <dat:list separator=",">
        <java:value-of select="sData">
          <data type="token"/>
        </java:value-of>
        <java:value-of select="iData">
          <data type="int"/>
        </java:value-of>
        <java:value-of select="dData">
          <data type="double"/>
        </java:value-of>
      </dat:list>
    </dat:byte>
  </define>

</grammar>
```

1.6. 6. define/ref

"resolveRef.xsl" ##### define/ref ######## XML ##### XSLT ##############

```
  [######]
```

Tips

```
src/xslt/raw/resolveRef.xsl
```