

#####

Table of contents

1 #####	2
2 1. #####	2
3 2. mapping definition ###	2
3.1 2.1 #####	3
3.2 2.2 #####	3
3.3 2.3 #####	5
3.4 2.4 #####	6
3.5 2.5 #####	7
4 3. #####	9
4.1 3.1 #####	9
4.2 3.2 #####	10
5 4. umsCodeGenerator ###	12
6 5. #####	12

#####

1.

umsCodeGenerator ##### CSV
#####

1. #####umsCodeGenerator #####
2. ##### mapping definition #####
3. ##### mapping definition #####
4. ##### umsCodeGenerator #####

2###3##### umsCodeGenerator #####
#####

#####umsCodeGenerator #####

umsCodeGenerator/yyyymmddvv/sample/tutorial #####
#####Java ## XML syntax#Language syntax#C ## XML syntax#Language syntax
#####

2. 1.

#####", " ##### CSV ##### "sample.csv"
#####

```
A,100,1.1
B,200,2.2
C,300,3.3
```

mappingSchema # #####(#####)#
#####(#### 1#4)# ##### mappingSchema #####(#### 2#3)#####

3. 2. mapping definition

mapping definition ##### mapping definition ##XML syntax#Language
syntax#C ##Java ##### 2.1 #####Java ## XML syntax #
mapping definition #####

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

#####

Note:
XML syntax # mapping definition # XML ## ### #####XML##### Emacs + nxml-mode#####

3.1. 2.1

#####"Sample" ##### (#####C #####)
#####

```
class Sample {
:
}
```

mapping definition # XML

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

Warning:
mapping definition ##### "ums"#"UMS" #####umsCodeGenerator ##
mappingSchema #####

3.2. 2.2

#####"decode"#####"encode"#####

#####(#####)#umsCodeGenerator ##### (mappingSchema
#####)#

##	###	#####
C##	#####	/** * ##### * * @param ums__buffer ##### * @param ums__bitlen #####(bit) * ##### * @param ums__ex #### *

#####

		@return #####(bit) */ ums__bitlen_t decode(ums__bitdata_t *ums__buffer, ums__bitlen_t ums__bitlen, ums__exception_t *ums__ex)
	#####	/** * ##### * * @param ums__buffer ##### * @param ums__bitlen #####(bit) * @param ums__ex ##### * @return #####(bit) */ ums__bitlen_t encode(ums__bitdata_t *ums__buffer, ums__bitlen_t ums__bitlen, ums__exception_t *ums__ex)
Java	#####	/** * ##### * * @param ums__buffer ##### * @param ums__bitlen #####(bit) * ##### * @return #####(bit) * @exception UMSEException ##### */ int decode(byte[] ums__buffer, int ums__bitlen) throws UMSEException
	#####	/** * ##### * * @param ums__buffer ##### * @param ums__bitlen #####(bit) * @return #####(bit) * @exception UMSEException ##### */ int encode(byte[] ums__buffer, int ums__bitlen) throws UMSEException

#####

```
class Sample {
    :
    int decode( byte[] ums__buffer, int ums__bitlen ) throws UMSEException {
    :
    }
    int encode( byte[] ums__buffer, int ums__bitlen ) throws UMSEException {
    :
    }
```

#####

```
}
}
```

mapping definition # XML

```
<?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">
      :
      <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="int"/>
          <java:exception type="UMSException"/>
          :
        </java:function>

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

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

3.3. 2.3 #####

mappingSchema ##### XML

A,100,1.1

", " ##### ##"#####"#####"#####"#####

```
<byte>
  <list separator=",">
    <data type="token"/>
    <data type="int"/>
    <data type="double"/>
  </list>
</byte>
```

#####

3.4.2.4

#####

```
class Sample {  
    String sData;  
    int    iData;  
    double dData;  
  
    int decode( byte[] ums__buffer, int ums__bitlen ) throws UMSEException {  
        :  
    }  
  
    int encode( byte[] ums__buffer, int ums__bitlen ) throws UMSEException {  
        :  
    }  
}
```

mappingSchema # XML

```
<?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="int"    />  
                    <java:exception type="UMSEException"/>  
                    :  
                </java:function>  
  
                <java:function name="encode">  
                    <java:arg      type="byte[]" name="ums__buffer"  
direction="out"/>  
                    <java:arg      type="int"    name="ums__bitlen" direction="in"/>  
                    <java:return   type="int"    />  
                    <java:exception type="UMSEException"/>  
                </java:function>  
            </defineFunctions>  
        </java:class>  
    </start>  
</grammar>
```

#####

```
        :
        </java:function>
    </defineFunctions>
</java:class>
</start>
</grammar>
```

3.5. 2.5

#####

```
class Sample {
    String sData;
    int    iData;
    double dData;

    int decode( byte[] ums__buffer, int ums__bitlen ) throws UMSEException {
        :
        sData = ... ; // 1#####
        :
        iData = ... ; // 2#####
        :
        dData = ... ; // 3#####
        :
    }

    int encode( byte[] ums__buffer, int ums__bitlen ) throws UMSEException {
        :
        ... = sData ; // 1#####
        :
        ... = iData ; // 2#####
        :
        ... = dData ; // 3#####
        :
    }
}
```

mapping definition # XML

```
<?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>
        </java:class>
    </start>
</grammar>
```

#####

```
</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="int"/>
    <java:exception type="UMSEException"/>

    <defineMapping direction="decode">
      <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>
    </defineMapping>
  </java:function>

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

    <defineMapping direction="encode">
      <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>
    </defineMapping>
  </java:function>

</defineFunctions>
</java:class>
```


#####

```
</start>
</grammar>
```

```
### mapping definition ##### ## mapping definition ##Sample.ums
#####
```

4.3.

```
##### main #####
```

1. #####
2. #####
3. #####
4. #####
5. #####

4.1. 3.1

```
#####
```

- ##### include(C ##) / import(java)
- #####
- #####

```
#####
```

##	#####	#####	####
C##	ums.h	tableTools_init()	tableTools_end()
Java	jp.jaxa.isas.ums.runtime.*	UMSLibrary.tableTools_	runUMSLibrary.tableTools_end()

```
main #####
```

```
import java.io.*;
import jp.jaxa.isas.ums.runtime.*;

class Main {

    private static final int BUFFER_SIZE = 4096;

    public static void main( String[] args ) {

        String inString      = null;
        byte[] inBuffer      = null;
        int    inBitlen      = 0;
        int    decodeBitlen  = 0;

        byte[] outBuffer     = new byte[BUFFER_SIZE];
        int    outBitlen     = 0;
```

#####

```
int    encodeBitlen = 0;

BufferedReader br    = null;

UMSLibrary.tableTools_init();

try {

    Sample sample = new Sample();
    br = new BufferedReader( new FileReader( args[0] ) );

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

        inBitlen = inString.length() * 8;
        inBuffer = inString.getBytes( "US-ASCII" );
        System.out.println(
            "input(" + inBitlen/8 + "*8+" + inBitlen%8 + "):<" + inString +
">" );

        /* call decode method */
        decodeBitlen = sample.decode( inBuffer, inBitlen );

        outBitlen = BUFFER_SIZE * 8;

        /* call encode method */
        encodeBitlen = sample.encode( outBuffer, outBitlen );

        String outString = new String( outBuffer, 0, encodeBitlen/8,
"US-ASCII");
        System.out.println(
            "output(" + encodeBitlen/8 + "*8+" + encodeBitlen%8 + "):<" +
outString.trim() + ">" );
    }

    } catch ( IOException ex ) {
        ex.printStackTrace( System.err );
    }

    } catch ( Throwable th ) {
        th.printStackTrace( System.err );
    }

    } finally {
        try {
            if ( br != null ) {
                br.close();
            }
        } catch ( Exception ex ) {}
    }

    UMSLibrary.tableTools_end();
}
}
```

4.2.3.2

#####

#####

- ##### include(C ##) / import(java)
- #####(C####)
- #####(C####)
- #####

#####

##	####	#####	#####	#####
C##	umsException.h	ums__exception_t	ums__exception_init	ums__exception_print
Java	jp.jaxa.isas.ums.runtime	UmsException	##	print

main #####

```
import java.io.*;
import jp.jaxa.isas.ums.runtime.*;

class Main {

    private static final int BUFFER_SIZE = 4096;

    public static void main( String[] args ) {

        String inString      = null;
        byte[] inBuffer      = null;
        int    inBitlen      = 0;
        int    decodeBitlen  = 0;

        byte[] outBuffer     = new byte[BUFFER_SIZE];
        int    outBitlen     = 0;
        int    encodeBitlen  = 0;

        BufferedReader br    = null;

        UMSLibrary.tableTools_init();

        try {

            Sample sample = new Sample();
            br = new BufferedReader( new FileReader( args[0] ) );

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

                inBitlen = inString.length() * 8;
                inBuffer = inString.getBytes( "US-ASCII" );
                System.out.println(
                    "input(" + inBitlen/8 + "*8+" + inBitlen%8 + "):<" + inString +
                    ">" );
            }
        }
    }
}
```

#####

```
/* call decode method */
decodeBitlen = sample.decode( inBuffer, inBitlen );

outBitlen = BUFFER_SIZE * 8;

/* call encode method */
encodeBitlen = sample.encode( outBuffer, outBitlen );

String outString = new String( outBuffer, 0, encodeBitlen/8,
"US-ASCII");
System.out.println(
    "output(" + encodeBitlen/8 + "*8+ " + encodeBitlen%8 + "):<" +
outString.trim() + ">" );
}

} catch ( UMSException ex ) {
ex.print( outBuffer, outBitlen );
ex.printStackTrace( System.err );

} catch ( IOException ex ) {
ex.printStackTrace( System.err );

} catch ( Throwable th ) {
th.printStackTrace( System.err );

} finally {
try {
if ( br != null ) {
br.close();
}
} catch ( Exception ex ) {}
}

UMSLibrary.tableTools_end();
}
}
```

main #####"Main.java"

5.4. umsCodeGenerator

mapping definition ## ##### umsCodeGenerator #####

#####"Sample.java" #####

```
$ txt2java -x Sample.ums
```

Warning:

```
#####mapping definition #####umsCodeGenerator ##### mapping definition
#####umsCodeGenerator #####
```

#####

6.5.

#####

#####(sample.csv#Main.java#Sample.java)#####

#####

- C##

```
$ gcc -std=c99 -ggdb -o main -Wall  
-I${TABLETOOLS_HOME}/build/include -I. *.c  
-L${TABLETOOLS_HOME}/build/lib/ -lums -lumstt  
$ ./main < sample.csv
```

- Java

```
$ javac -classpath .:${TABLETOOLS_HOME}/build/classes  
Main.java Sample.java  
$ java -cp .:${TABLETOOLS_HOME}/build/classes Main sample.csv
```

#####

```
input(9*8+0):<A,100,1.1>  
output(9*8+0):<A,100,1.1>  
input(9*8+0):<B,200,2.2>  
output(9*8+0):<B,200,2.2>  
input(9*8+0):<C,300,3.3>  
output(9*8+0):<C,300,3.3>
```