

umsCodeGenerator

Table of contents

1 mappingSchema.....	2
1.1 Page for mappingSchema in general.....	2
1.2 Specification of mapping definition.....	4
1.3 userGuide.....	10
2 umsCodeGenerator.....	79
2.1 umsCodeGenerator.....	79
2.2 umsCodeGenerator #####.....	82
2.3 umsCodeGenerator ####.....	83
2.4 #####.....	84
2.5 Tips.....	96
2.6 umsCodeGenerator 0.4.....	101
3 Developer's Inforamtion.....	105
3.1 umsCodeGenerator.....	105
3.2 Todo List.....	128
3.3 Related Information for usage and development of mappingSchema.....	129
3.4 Others.....	130
4 All.....	132

1. mappingSchema

1.1. Page for mappingSchema in general

[English](#)

1.1.1. mappingSchema

```
mappingSchema #####(#####) #####  
##### XML ##### RELAX NG  
#####
```

```
#####RELAX NG ##### UMS #### ##### RELAX NG  
# UMS #####
```

- [RELAX NG](http://www.asahi-net.or.jp/~eb2m-mrt/relaxngjis/jaspec-20011203.html) (http://www.asahi-net.or.jp/~eb2m-mrt/relaxngjis/jaspec-20011203.html)
- UMS ### (2006/07/27) ([pdf](#))

1.1.2. UMS

```
UMS(Universal Mapping Schema)#####  
#####  
#####  
#####  
#####  
#####  
#####
```

1.1.3. mapping definition

```
mappingSchema ##### mapping definition##### mapping definition##XML  
##### XML ##### mappingSchema #####  
(umsCodeGenerator #####) # XML ##### (RELAX NG)  
#####
```

```
mapping definition # #####(#####)# #####  
mapping definition# #####  
#####  
#####  
##### mapping definition#####
```

```
mapping definition#####  
#####umsCodegenerator #####  
mappingSchema ##### mappingSchema #####  
mapping definition## ##### mapping definition#  
##### ## mapping definition (#####)  
#####
```

1.1.4.

1.1.4.1. Required knowledge

Knowledge of XML and its name space is required for usage of mappingSchema since mappingSchema is a schema language described in XML (sorry, no description is available at this moment). mappingSchema shares philosophy and some definition with an XML schema language [RELAX NG](http://www.thaiopensource.com/relaxng/) (http://www.thaiopensource.com/relaxng/) . Knowledge of RELAX NG is helpful to understand mappingSchema. Although XML version of mappingSchema is very similar with that of RELAX NG, there is some difference (see [difference](#)).

1.1.4.2. How To

In trodution of software development with mappingSchema is found [here](#) (sorry in Japanese).

develop program with mappingSchema (sorry, in Japanese). Note: Mapping definition and tools used in this document is somewhat older than the recent implementation of umsCodeGenerator.

1.1.4.3. Development

is now on-going by myself (KM) and few members. To accelerate our development, any contribution or collaboration is welcome.

1.1.4.4. Mailing Lists

Please send mail Mailing List itself if you want to join.

- Users ML: ums_users@plain.isas.jaxa.jp
- Develloppers ML: ums_devel@plain.isas.jaxa.jp

1.1.5.

1.1.5.1. #####

- ##### API
#####
- ##### (ums group) #####
- #####
- #####
- Encode # decode ##### XML - XML syntax ##### XML - language syntax
#####

(-KM-)

1.1.5.2. PDF#####

[Apache Forrest](http://forrest.apache.org/) (http://forrest.apache.org/) #####
PDF#####

- #####?
- #####
- #####

1.2. Specification of mapping definition

English

1.2.1. 1. mapping definition ###

mapping definition # #####

- #####
- #####
- #####(#####)

mappingSchema #0.4#### ###URI##### ## #####
#####

##	#### URI	#####
#####	http://ums.isas.jaxa.jp/0.4/	dat
	C##	ching
#####	Java	java
	Perl	perl
#####	http://ums.isas.jaxa.jp/0.4/	(#####)

Note:

XML ###

1.2.1.1. 1.1 #####

byte ### bit ### list ##### data ##### value

#####1##### ##### XML ## # #####

```
<dat:byte enocde="signed" length="1">
  <dat:data type="byte"/>
</dat:byte>
```

1.2.1.2. 1.2 #####

syntax

- XML-XML syntax
- XML-language syntax

XML-XML syntax # XML-language syntax

1.2.1 #####(XML syntax)

XML syntax ##### 'lang:value-of'## ##### 'lang:data'## ##
'lang:value'## #####

#####1##### var ##### ##### XML ## # #####

```
<lang:value-of select="var">
  <lang:data type="byte"/>
</lang:value-of>
```

1.2.2 #####(Language syntax)

Language syntax ##### ##### 'lang:data'## ## 'lang:value'##
#####

#####1##### var ##### ##### XML ## #XML #####
#####

```
var = <lang:data type="byte"/>;
<lang:data type="byte"/> = var;
```

1.2.1.3. 1.3 #####

#####

1.3.1 #####(XML syntax)

XML syntax ##### byte ### bit ### list #####
 lang:value-of ## ##### lang:data ## ### lang:value ## #####

#####1##### var #1##### ##### dat:byte
 ## # ##### lang:value-of ## # ##### data ## # #####

```
<dat:byte enocde="signed" length="1">
  <lang:value-of select="a">
    <data type="byte"/>
  </lang:value-of>
</dat:byte>
```

XML syntax #####/##### mapping definition #####

##	data	value
#####	<dat:byte enocde="signed" length="1"> <dat:data type="byte"/> </dat:byte>	<dat:byte enocde="signed" length="1"> <dat:value type="byte">1</dat:value> </dat:byte>
#####	<lang:value-of select="var"> <lang:data type="byte"/> </lang:value-of>	<lang:value-of select="var"> <lang:value type="byte">1</lang:value> </lang:value-of>
#####	<dat:byte enocde="signed" length="1"> <lang:value-of select="a"> <data type="byte"/> </lang:value-of> </dat:byte>	<dat:byte enocde="signed" length="1"> <lang:value-of select="a"> <value type="byte"></value> </lang:value-of> </dat:byte>

Note:
 ##### #XSL##### XML-language syntax ### XML-XML syntax ##### XML-XML syntax # ##### #
 XML-language syntax #####

1.3.2 #####(Language syntax)

Language syntax ##### byte ### bit ### list #####
lang:data ## ### lang:value ##

#####1##### var #1#####
dat:byte ## # ##### data ## # #####

```
<dat:byte enocde="signed" length="1">
  a = <data type="byte"/>;
</dat:byte>
```

```
<dat:byte enocde="signed" length="1">
  <data type="byte"/> = a;
</dat:byte>
```

Language syntax ##### mapping definition #####

##	data	value
#####	<dat:byte enocde="signed" length="1"> <dat:data type="byte"/> </dat:byte>	<dat:byte enocde="signed" length="1"> <dat:value type="byte">1</dat:value> </dat:byte>
#####	<lang:data type="byte"/> = a;	<lang:value type="byte">1</lang:value> = a;
#####	<dat:byte enocde="signed" length="1"> <data type="byte"/> = a; </dat:byte>	<dat:byte enocde="signed" length="1"> <value type="byte">1</value> = a; </dat:byte>

mapping definition

##	data	value
#####	<dat:byte enocde="signed" length="1"> <dat:data type="byte"/>	<dat:byte enocde="signed" length="1"> <dat:value type="byte">1</dat:value>

	</dat:byte>	</dat:byte>
#####	a = <lang:data type="byte" />;	<lang:value type="byte">1</lang:value> = a;
####	<dat:byte enocde="signed" length="1"> a = <data type="byte" />; </dat:byte>	<dat:byte enocde="signed" length="1"> a = <value type="byte">1</value>; </dat:byte>

Note:
 ##### XML-XML syntax ### XML-language syntax ### #####
 XML-language syntax ##### ##### XML-language syntax # ##### ##### XML-XML
 syntax ##### #####

1.2.2. 2. RELAX NG #####? - 2005/06/28##

RELAX NG # XML ### "##" ##### mappingSchema #####
 #####
 #####"##" ##### -
 ##### - mappingSchema #####
 #####
 ##### "##" ##### - ##### - mappingSchema #####"##"
 #####mappingSchema ##### RELAX NG ##
 #####

1.2.2.1. 2.1 ## (#####) - 2005/07/21##

RELAX NG ##### (see RELAX NG 4.1)# mappingSchema
 #####

```
<element name="A">
  <other:comment>
    <element name="B">
      <text/>
    </element>
  </other:comment>
</element>
```


RELAX NG ##

```
<element name="A" />
```

mappingSchema

```
<element name="A">  
  <element name="B">  
    <text />  
  </element>  
</element>
```

#####

1.2.2.2. 2.2 ##### - 2005/08/01##

mappingSchema #####

```
<byte encode="txt">  
  <data type="int" />  
</byte>  
<byte encode="txt">  
  <data type="int" />  
</byte>
```

'123456' ##### 2 ##### 5
#####

- '12345' ## '6'
- '1234' ## '56'
- '123' ## '456'
- '12' ## '3456'
- '1' ## '23456'

#####

RELAX NG (see 7.2) ## mappingSchema ##### (list #####)#

```
<data type="int" />  
<data type="int" />
```

1.2.3. 3. #####

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

- optional ###oneOrMore ###zeroOrMore ##### 32 ###
- list ##### 32 ###

- define/ref #####
- define #####optional #oneOrMore#zeroOrMore #####
- dat:*//ums:* #####
- XML syntax #####/#####
- Language syntax #####XML syntax #
 <package>#<class>#<function>#<arg>#<return>#<var>#<array>#
 <value-of>#<callFunction>##### \$TABLETOOLS_HOME/sample/tutorial #####
 Language syntax #####

1.3. userGuide

1.3.1. grammar

1.3.1.1.

- ### ##### start##, define##, include##, ref##
#####
- ##### ### ### ##### ref##
#####(###externalRef##
#####)#
- XML##### ### # element## #####

1.3.1.2.

xmlns ##

#####

RELAX NG ##### URI ##http://relaxng.org/ns/structure/1.0 ###

ns ##

#####

xmlns:xxxx ##

#####(xxxx
#####)##### xmlns:xxxx ## #####ns
#####

1.3.1.3.

- ##### start## #####

1.3.1.4.

1:

```
<?xml version="1.0"?>
<grammar xmlns="http://relaxng.org/ns/structure/1.0"
  ns="http://lena.org/" xmlns:h="http://batz.org/">

<start>
  <element name="document">
    <ref name="para"/>
  </element>
</start>

<define name="para">
  <element name="h:para">
    <empty/>
  </element>
</define>

</grammar>
```

1.3.1.5.

- James Clark and Makoto Murata, ISO/IEC FDIS 19757-2 Document Schema Definition Language (DSDL) -- Part 2: Regular-grammar-based validation -- RELAX NG
- ## #, #grammar## @ ##### RELAX NG ###

1.3.2. start

1.3.2.1. XML

- #####(XML Schema # DTD
#####)#

1.3.2.2.

combine ##

```
### grammar## ##### start## ##### choice #####
interleave ##### start## ##### combine ## ##### (#####)
##### XML ##### interleave #####
grammar## #####
```

1.3.2.3.

- `### ### ### ### ### #####`
- `ref## #####(#####)#`
`#####XML ##### define## ##### element## #####`
- `XML #####(#####)##### grammar## ##### element##`
`#####choice##`
`#####`

1.3.2.4. ###

1: ref##

```
<?xml version="1.0"?>
<grammar xmlns="http://relaxng.org/ns/structure/1.0"
  ns="http://lena.org/">

<start>
  <ref name="document"/>
</start>

<define name="document">
  <element name="document">
    <empty/>
  </element>
</define>
</grammar>
```

2: choice##

`##### doc1, doc2 #####`

```
<start>
  <choice>
    <element name="doc1">
      <text/>
    </element>
    <element name="doc2">
      <empty/>
    </element>
  </choice>
</start>
```

3: start##

`#####`

```
<start>
  <element name="doc1">
    <text/>
```

```
</element>
</start>

<start combine="choice">
  <element name="doc2">
    <empty/>
  </element>
</start>
```

1.3.2.5. #####

- James Clark and Makoto Murata, ISO/IEC FDIS 19757-2 Document Schema Definition Language (DSDL) -- Part 2: Regular-grammar-based validation -- RELAX NG
- ## #, #start## @ ##### RELAX NG ###

1.3.3. file ##

1.3.3.1. #####

- ##### basename #####
- ums ##### basename # basename #####
- grammar ##### #####

1.3.3.2. ##

basename ##

#####

1.3.3.3. #####

- ### ##### #####

1.3.3.4. ###

1: 1#####

FileSample.ums #####FileSample.java #File.java

```
<grammar xmlns="http://ums.isas.jaxa.jp/0.4"
datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes">
  <start>
    <file basename="File">
      <java:class scope="abstract" name="FileSample"
xmlns:java="http://ums.isas.jaxa.jp/0.4/java">
        ...
      </java:class>
      <java:class name="FileSampleRecord"
```

```

xmlns:java="http://ums.isas.jaxa.jp/0.4/java">
    ...
    </java:class>
  </file>
</start>
</grammar>

```

2:

**### FileSample.ums #####FileSample.java #File.java #FileRecord.java
#####**

```

<grammar xmlns="http://ums.isas.jaxa.jp/0.4"
datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes">
  <start>
    <file basename="File">
      <java:class scope="abstract" name="FileSample"
xmlns:java="http://ums.isas.jaxa.jp/0.4/java">
        ...
        </java:class>
      </file>
      <file basename="FileRecord">
        <java:class name="FileSampleRecord"
xmlns:java="http://ums.isas.jaxa.jp/0.4/java">
          ...
          </java:class>
        </file>
      </start>
    </grammar>

```

3:

**### FileSample.ums #####FileSample.java #File.java #FileRecord.java
#####**

```

<grammar xmlns="http://ums.isas.jaxa.jp/0.4"
datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes">
  <start>
    <file basename="File">
      <file basename="File">
        <java:class scope="abstract" name="FileSample"
xmlns:java="http://ums.isas.jaxa.jp/0.4/java">
          ...
          </java:class>
        </file>
        <file basename="FileRecord">
          <java:class name="FileSampleRecord"
xmlns:java="http://ums.isas.jaxa.jp/0.4/java">
            ...
            </java:class>
          </file>
        </start>
      </grammar>

```

```
        ...
        </java:class>
    </file>
</file>
</start>
</grammar>
```

1.3.3.5.

#####ums ##### basename # basename #####

1.3.4. include

1.3.4.1.

- #####
- ##### grammar##
externalRef##
- ### grammar## #####
- ref## #####
- #####

1.3.4.2.

href ##

- ##### ## URI ###

1.3.4.3.

- #####
- #####start##, div##, define## #####
- include## ### define## ## include##
#####combine ##
#####

1.3.4.4.

1:

#####

#####: schema1.rng

```

<?xml version="1.0"?>
<grammar xmlns="http://relaxng.org/ns/structure/1.0"
  ns="http://lena.org/">

<start>
  <element name="document">
    <ref name="para"/>
  </element>
</start>

<include href="schema2.rng">
  <define name="em-content"> <!-- "em-content" ### -->
    <text/>
  </define>
</include>

</grammar>

```

####2: schema2.rng

```

<?xml version="1.0"?>
<grammar xmlns="http://relaxng.org/ns/structure/1.0"
  ns="http://lena.org/">

<define name="para">
  <element name="em">
    <ref name="em-content"/>
  </element>
</define>

<define name="em-content">
  <empty/>
</define>

</grammar>

```

schemal.rng ##### (#####)#

```

<?xml version="1.0"?>
<grammar xmlns="http://relaxng.org/ns/structure/1.0"
  ns="http://lena.org/">

<start>
  <element name="document">
    <ref name="para"/>
  </element>
</start>

<define name="para">
  <element name="em">
    <ref name="em-content"/>
  </element>
</define>

```



```
<define name="em-content">
  <text/> <!-- define ##### -->
</define>

</grammar>
```

2:

#####

####1: schema1.rng

```
<?xml version="1.0"?>
<grammar xmlns="http://relaxng.org/ns/structure/1.0"
  ns="http://lena.org/">

<start>
  <element name="document">
    <ref name="para"/>
  </element>
</start>

<include href="schema2.rng"/>

</grammar>
```

####2: schema2.rng

```
<?xml version="1.0"?>
<grammar xmlns="http://relaxng.org/ns/structure/1.0"
  ns="http://batz.org/">

<define name="para">
  <element name="em"><empty/></element>
</define>

</grammar>
```

schema1.rng

```
<?xml version="1.0"?>
<grammar xmlns="http://relaxng.org/ns/structure/1.0"
  xmlns:lena="http://lena.org/" xmlns:batz="http://batz.org/">

<start>
  <element name="lena:document">
    <ref name="para"/>
  </element>
</start>

<define name="para">
  <element name="batz:em"><empty/></element>
</define>
```

</grammar>

1.3.4.5. #####

- James Clark and Makoto Murata, ISO/IEC FDIS 19757-2 Document Schema Definition Language (DSDL) -- Part 2: Regular-grammar-based validation -- RELAX NG
- ## #, #include## @ ##### RELAX NG ###

1.3.5. define ##

1.3.5.1. #####

- #####(#####)##### define##
#####
 - define## ##### ref## ##### externalRef##, include## #####
 - ## grammar## #### include## ##### define##
#####
 - ##### define## # ref## ##### ##include##
#####(ref##) ##(define##)
ref##
- ##### define##
- grammar## #####combine ## #####combine ## ##
choice ##### interleave #####
 - include## #####
#####combine ## #####

1.3.5.2. ##

name ##

#####name ## ##### NCName
:
#####

combine ##

#####(#####)#

1.3.5.3. #####

- ## define##
#####empty##
notAllowed##

- notAllowed## #####)#
- ##### element##, attribute##
#####(#####)#####
define, ref #####

```
<element>
  <anyName>
    <except> <!-- except ##### -->
      <ref name="exceptRule"/> <!-- #####!! -->
    </except>
  </anyName>
  ...
</element>

<define name="exceptRule">
  <choice>
    <name>abc</name>
    <name>def</name>
  </choice>
</define>
```

#####ref##, define## ##### define## #### except## #####

1.3.5.4.

1: define## # include##

####1: schema1.rng

```
<?xml version="1.0"?>
<grammar xmlns="http://relaxng.org/ns/structure/1.0">

<start>
  <element name="document">
    <ref name="para"/> <!-- ##### -->
  </element>
</start>

<include href="schema2.rng"/> <!-- ##### -->

<define name="para" combine="choice">
  <element name="strong"><empty/></element>
</define>

</grammar>
```

####2: schema2.rng

```
<?xml version="1.0"?>
<grammar xmlns="http://relaxng.org/ns/structure/1.0">

<define name="para"> <!-- ##### define ## -->
```

```
<element name="em"><empty/></element>
</define>

</grammar>
```

2:

#####

```
<?xml version="1.0"?>
<grammar xmlns="http://relaxng.org/ns/structure/1.0">

<start>
  <element name="document">
    <choice> <!-- define ### combine ##### -->
      <element name="strong"><empty/></element>
      <element name="em"><empty/></element>
    </choice>
  </element>
</start>

</grammar>
```

1.3.5.5. ####

- James Clark and Makoto Murata, ISO/IEC FDIS 19757-2 Document Schema Definition Language (DSDL) -- Part 2: Regular-grammar-based validation -- RELAX NG
- ## #, #define## @ #### RELAX NG ###

1.3.6. ref ##

1.3.6.1. #####

- define## #####
 - include## # externalRef## ##### ref## #
define## #####
 - ##### (XML #####) #####
#####XML # name##, anyName##, nsName##
#####
 - #####
- ##### nest
#####

```
<define name="NestPattern">
  <element name="nest">
    <ref name="NestPattern"/>
  </element>
```

```
</define>
```

XML

```
#####
```

- ##### externalRef
#####

1.3.6.2.

name ##

```
##### NCName
```

```
#####
```

```
define## #####
```

ns,xmlns:xxx,datatypeLibrary ##

```
#### ns ## ,xmlns:xxxx ## ,datatypeLibrary ##
```

```
#####
```

1.3.6.3.

- #####

1.3.6.4.

1:

```
define## #####
```

1.3.6.5.

- James Clark and Makoto Murata, ISO/IEC FDIS 19757-2 Document Schema Definition Language (DSDL) -- Part 2: Regular-grammar-based validation -- RELAX NG
- ## #, #ref## @ #### RELAX NG ###

1.3.7. externalRef

1.3.7.1.

- #####
- include## # grammar## #####externalRef##
#####
- ##### (XML #####) #####

```
####XML # name##, anyName##, nsName##
#####
• #####
• ref## #####define##
#####externalRef##
#####define## #####
• grammar #####
```

1.3.7.2.

href ##

- ##### ## URI ###

ns,xmlns:xxx,datatypeLibrary ##

```
#### ns ## ,xmlns:xxxx ## , datatypeLibrary ##
#####(#####)#
```

1.3.7.3.

- #####

1.3.7.4.

1: externalRef##

schema1.rng

```
<?xml version="1.0"?>
<grammar xmlns="http://relaxng.org/ns/structure/1.0">

<start>
  <element name="drinkshop">
    <externalRef href="schema2.rng"/> <!-- ##### -->
  </element>
</start>

</grammar>
```

schema2.rng (##### ; #### choice ##)

```
<?xml version="1.0"?>
<choice xmlns="http://relaxng.org/ns/structure/1.0">
  <element name="favoriteItem">
    <text/>
  </element>
```

```
<element name="regularItem">  
  <text/>  
</element>  
</choice>
```

2: externalRef

schemal.rng ##### (##### externalRef #####)#

```
<?xml version="1.0"?>  
<grammar xmlns="http://relaxng.org/ns/structure/1.0">  
  
<start>  
  <element name="drinkshop">  
    <choice>  
      <element name="favoriteItem">  
        <text/>  
      </element>  
      <element name="regularItem">  
        <text/>  
      </element>  
    </choice>  
  </element>  
</start>  
  
</grammar>
```

1.3.7.5. ####

- James Clark and Makoto Murata, ISO/IEC FDIS 19757-2 Document Schema Definition Language (DSDL) -- Part 2: Regular-grammar-based validation -- RELAX NG
- ## #, #externalRef## @ ##### RELAX NG ###

1.3.8. parentRef ##

1.3.8.1. #####

- ##### ref## #####grammar## #####
 grammar## #####
- parentRef## #####
- grammar##
 #####

1.3.8.2. ##

name ##

#####

1.3.8.3. #####

(ref## ###)

- #####

1.3.8.4. ###

1: ref##

#####

```
<grammar xmlns="http://relaxng.org/ns/structure/1.0">
<define name="#####1">
  <value>#####</value>
</define>
<start>
  <element name="#">
    <choice>
      <ref name="#####1"/>
      <grammar>
        <start>
          <element name="#">
            <ref name="#####2"/>
          </element>
        </start>
        <define name="#####2">
          <value>#####</value>
        </define>
      </grammar>
    </choice>
  </element>
</start>
</grammar>
```

grammar## #####1#####

ref## ##### grammar##

#####

- <#>#####</#>
- <#><#>#####</#></#>

###ref## ##### #1 #####2 #####

2: parentRef##

#####1#####

```
<grammar xmlns="http://relaxng.org/ns/structure/1.0">
<define name="####1">
  <value>####</value>
</define>
<start>
  <element name="#">
    <choice>
      <ref name="####1"/>
      <grammar>
        <start>
          <element name="#">
            <parentRef name="####1"/>
          </element>
        </start>
        <define name="####2">
          <value>#####</value>
        </define>
      </grammar>
    </choice>
  </element>
</start>
</grammar>
```

#####1#####

1.3.8.5. ####

- James Clark and Makoto Murata, ISO/IEC FDIS 19757-2 Document Schema Definition Language (DSDL) -- Part 2: Regular-grammar-based validation -- RELAX NG
- ## #, #parentRef## @ ##### RELAX NG ###

1.3.9. defineFuncitons ##

1.3.9.1. #####

- #####

1.3.9.2. ##

- #####

1.3.9.3. ##

- start ###class ###java #####

1.3.9.4. #####

- function #####

1.3.9.5. ###

1:

```
<defineFunctions>
  <lang:function name="setValue">
    <lang:arg type="int" name="iValue" direction="in"/>
    <lang:return type="void"/>
  </lang:function>
  <lang:function name="getValue">
    <lang:return type="int"/>
  </lang:function>
</defineFunctions>
```

1.3.10. defineMapping ##

1.3.10.1. #####

- #####
- #####

1.3.10.2. ##

direction ##

#####

- decode : #####
- encode : #####

1.3.10.3. ##

- function #####

1.3.10.4. ####

- ### #### ##### #####
- interleave##, optional##, zeroOrMore## #####

1.3.10.5. ###

1: decode

umsCodeGenerator

```
<defineMapping direction="decode"
xmlns:txt="http://ums.isas.jaxa.jp/0.4/dat">
  <txt:list encode="txt">
    <txt:data type="token"/>
    <lang:value-of select="strData">
      <data type="token"/>
    </lang:value-of>
    <lang:callFunction expr="setStringValue( strData );"/>
    <lang:value-of select="iData">
      <data type="int"/>
    </lang:value-of>
    <lang:callFunction expr="setIntValue( iData );"/>
  </txt:list>
</defineMapping>
```

2: encode

```
<defineMapping direction="encode"
xmlns:txt="http://ums.isas.jaxa.jp/0.4/dat">
  <txt:list encode="txt" delimiter=" ">
    <lang:callFunction expr="strData = getStringValue();"/>
    <lang:value-of select="strData">
      <data type="token"/>
    </lang:value-of>

    <lang:callFunction expr="iData = getIntValue();"/>
    <lang:value-of select="iData">
      <lang:data type="int"/>
    </lang:value-of>
  </txt:list>
</defineMapping>
```

1.3.11. defineVariables

1.3.11.1.

- #####
- start ##### fuction #####

1.3.11.2.

- #####

1.3.11.3.

- start ###class ###java #####function #####

1.3.11.4.

- **### array ###var #####**

1.3.11.5.

1:

```
<defineVariables>
  <lang:array size="4096">
    <lang:var type="char" name="strData"/>
  </lang:array>
  <lang:var type="int" name="iData"/>
</defineVariables>
```

1.3.12. group

1.3.12.1.

- ##### group## #####
- ##### # ##### (XML#####) #####
- grammar #####
- ##### externalRef
#####

1.3.12.2.

- #####

1.3.12.3.

- #####choice## #####
- ##### # (XML#####) # #####
- optional##, choice##, zeroOrMore## #####
- group## #####
- ref #####

1.3.12.4.

1:

```
<?xml version="1.0"?>
<element name="start" xmlns="http://relaxng.org/ns/structure/1.0">
  <choice>
    <group>
      <attribute name="myway">
        <value>1</value>
```

```
</attribute>
<element name="Japan-way"><empty/></element>
</group>
<group>
  <attribute name="myway">
    <value>2</value>
  </attribute>
  <element name="China-way"><empty/></element>
</group>
</choice>
</element>
```

#####

- ```
<?xml version="1.0"?>
<start myway="1">
 <Japan-way/>
</start>
```
- ```
<?xml version="1.0"?>
<start myway="2">
  <China-way/>
</start>
```

#####

- ```
<?xml version="1.0"?>
<start myway="lena"> <!-- myway #####1#2## -->
 <Japan-way/>
</start>
```
- ```
<?xml version="1.0"?>
<start myway="1">
  <China-way/> <!-- myway="1" ##### Japan-way #### -->
</start>
```

1.3.12.5.

- James Clark and Makoto Murata, ISO/IEC FDIS 19757-2 Document Schema Definition Language (DSDL) -- Part 2: Regular-grammar-based validation -- RELAX NG
- ## #, #group## @ ##### RELAX NG ###

1.3.13. interleave

1.3.13.1.

- #####group##
#####interleave## #####
- XML#####attribute##
#####
- ##### # ##### (XML#####) #####

- grammar #####
- ##### externalRef
#####

1.3.13.2. ##

- #####

1.3.13.3. ##

interleave #####32####

1.3.13.4. ####

- ### #### #####
- choice##, optional##, zeroOrMore## #####
- interleave## ##### (#####)
#####
- ref #####

1.3.13.5. ###

1:

#####

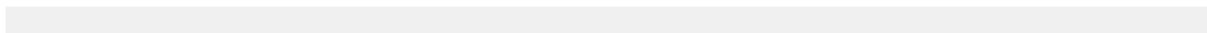
```
<element name="doc">
  <interleave>
    <element name="a"><empty/></element>
    <element name="b"><empty/></element>
    <element name="c"><empty/></element>
  </interleave>
</element>
```

#####

- <doc><a/><c/></doc>
- <doc><a/><c/></doc>
- <doc><a/><c/></doc>
- <doc><c/><a/></doc>
- <doc><c/><a/></doc>
- <doc><c/><a/></doc>

2: group # interleave

#####



```
<interleave>  
  <group>  
    <element name="a"><text/></element>  
    <element name="b"><text/></element>  
  </group>  
  <element name="c"><text/></element>  
</interleave>
```

#####

- <a/><c/>
- <a/><c/>
- <c/><a/>

1.3.13.6.

1: ##### interleave

interleave## ### interleave## #####

###1

```
<interleave>  
  <ref name="a"/>  
  <interleave>  
    <ref name="b"/>  
    <ref name="c"/>  
  </interleave>  
</interleave>
```

###2

```
<interleave>  
  <ref name="a"/>  
  <ref name="b"/>  
  <ref name="c"/>  
</interleave>
```

2: empty## ##### interleave##

interleave## #####1#### empty## ##### empty## #####

3: notAllowed## ##### interleave##

interleave## #####1#### notAllowed## #####interleave##
#####1## notAllowed## #####

#####

1. #####
2. ##### (##### interleave##) #####

3. #####
#####

1.3.13.7. ####

- James Clark and Makoto Murata, ISO/IEC FDIS 19757-2 Document Schema Definition Language (DSDL) -- Part 2: Regular-grammar-based validation -- RELAX NG
- ## #, #interleave## @ ##### RELAX NG ###

1.3.14. choice ##

1.3.14.1. #####

- #####
- ##### # ##### (XML#####) #####
- grammar #####
- ##### externalRef
#####

1.3.14.2. ##

- #####

1.3.14.3. ####

- ### ### #####
- interleave##, optional##, zeroOrMore## #####
- #####(#####)#####
- ref #####

1.3.14.4. ###

1: mapping ## (XML-XML syntax)

#####'A' ##### '2' ##### ## id ## '1'#'2'#####

```
<choice>
  <group>
    <dat:byte><dat:value>A</dat:value></dat:byte>
    <lang:value-of
select="id"><lang:value>1</lang:value></lang:value-of>
  </group>
  <group>
    <dat:byte>
      <lang:value-of select="id"><value>2</value></lang:value-of>
    </dat:byte>
  </group>
</choice>
```



```
</group>  
</choice>
```

2: mapping ## (XML-language syntax)

1

encode ##

```
<choice>  
  <group>  
    <lang:value>1</lang:value> = id;  
    <dat:byte><dat:value>A</dat:value></dat:byte>  
  </group>  
  <group>  
    <dat:byte>  
      <value>2</value> = id;  
    </dat:byte>  
  </group>  
</choice>
```

decode ##

```
<choice>  
  <group>  
    <dat:byte><dat:value>A</dat:value></dat:byte>  
    id = <lang:value>1</lang:value>;  
  </group>  
  <group>  
    <dat:byte>  
      id = <value>2</value>;  
    </dat:byte>  
  </group>  
</choice>
```

choice ### group #####
#####

1.3.14.5. #####

1: #### choice##

choice## ### choice## #####

###1

```
<choice>  
  <ref name="a" />  
  <choice>  
    <ref name="b" />
```

```
<ref name="c"/>
</choice>
</choice>
```

###2

```
<choice>
  <ref name="a"/>
  <ref name="b"/>
  <ref name="c"/>
</choice>
```

2: empty## #### choice##

```
#####empty## ##### 2##### empty##
#####notAllowed## ##### =
#####(#####)#
```

3: notAllowed## #### choice##

```
notAllowed## #####notAllowed##
#####
```

```
#####
```

1.3.14.6.

- James Clark and Makoto Murata, ISO/IEC FDIS 19757-2 Document Schema Definition Language (DSDL) -- Part 2: Regular-grammar-based validation -- RELAX NG
- ## #, #choice## @ #### RELAX NG ###

1.3.15. optional

1.3.15.1.

- #####0#####1#####
- #### attribute## #####(DTD ## #IMPLIED #####)#
- attribute## #####
- ##### # ##### (XML#####) #####
- grammar #####
- ##### externalRef #####

1.3.15.2.

lang:occurs ##

lang #####optional

- decode : #1#####optional #####
- encode : #1#####optional #####(encode #####)#

lang:occured ##

lang #####optional

- decode : optional #####1## #####0#####
- encode : #####

1.3.15.3. ##

- optional#oneOrMore#zeroOrMore #####32####
- ref #####define/ref #####
- lang:occurs ####lang:occured #####
- occurs ### 0 ### 1 #####

1.3.15.4. ####

- ### #####
- interleave##, optional##, zeroOrMore## #####

1.3.15.5. #####

1: #### optional##

optional## ##### optional##

#1

```
<optional>
  <ref name="a"/>
</optional>
<optional>
  <ref name="b"/>
</optional>
```

#2

```
<optional>
  <ref name="a"/>
  <ref name="b"/>
</optional>
```

#1####a#b#####2####a#b#####

optional## #####

2:

choice##,interleave## #####

```
<element name="p">
  <optional>
    <element name="a"><empty/></element>
  </optional>
  <optional>
    <element name="b"><empty/></element>
  </optional>
</element>
```

#####

```
<p><b/><a/></p>
```

#####(#####choice##,
interleave## #####)#

1.3.15.6. ####

- James Clark and Makoto Murata, ISO/IEC FDIS 19757-2 Document Schema Definition Language (DSDL) -- Part 2: Regular-grammar-based validation -- RELAX NG
- ## #, #optional## @ #### RELAX NG ###

1.3.16. oneOrMore ##

1.3.16.1. 1#####

- #####1#####
- XML #####
- attribute## #####(attribute## # element##
#####)#####
- attribute## #####
- grammar #####
- ##### externalRef
#####
- #####

1.3.16.2. ##

lang:occurs ##

lang

umsCodeGenerator

- decode : #####
- encode : #####(encode #####)#

lang:occured

lang #####oneOrMore

- decode : #####
- encode : #####

1.3.16.3.

- optional#oneOrMore#zeroOrMore #####32####
- ref #####define/ref #####
- lang:occurs ####lang:occured #####
- lang:occurs ### 1 #####

1.3.16.4.

- ### #####
- interleave##, optional##, zeroOrMore## #####

1.3.16.5.

1: mapping ## (XML-XML syntax)

#####1 ##### ## n #####

decode

```
<oneOrMore lang:occured="n">
  <dat:byte length="1">
    <lang:value-of select="i">
      <data type="int"/>
    </lang:value-of>
  </dat:byte>
  <callFunction expr="pushValue( i );"/>
</oneOrMore>
```

encode

```
<oneOrMore lang:occurs="n">
  <callFunction expr="i = popValue();"/>
  <dat:byte length="1">
    <lang:value-of select="i">
      <data type="int"/>
    </lang:value-of>
  </dat:byte>
</oneOrMore>
```

decode ##### n ##### encode ##### n #####

2: mapping ## (XML-language syntax)

1

decode

```
<oneOrMore>
  <dat:byte length="1">
    <callFunction expr="pushValue( <data type="int"/> );">
  </dat:byte>
  n =
</oneOrMore>;
```

encode

```
<oneOrMore>
  = n;
  <dat:byte length="1">
    <data type="int"/> = <callFunction expr="popValue();"/>
  </dat:byte>
</oneOrMore>
```

3: JAVA ## mapping ## (XML-language syntax)

JAVA ## Iterator # Collection #####

Iterator ##### encode

```
<oneOrMore>
  <lang:var class="Byte" name="b" iterator="iterator">
    <dat:byte length="1">
      <data type="byte"/> = b.byteValue();
    </dat:byte>
  </lang:var>
</oneOrMore>
```

List ##### encode

```
<oneOrMore>
  <lang:var class="Byte" name="b" collection="list">
    <dat:byte length="1">
      <data type="byte"/> = b.byteValue();
    </dat:byte>
  </lang:var>
</oneOrMore>
```

List ##### decode

```
<oneOrMore>  
  <dat:byte length="1">  
    list.add( new Byte(<data type="byte"/>) );  
  </dat:byte>  
</oneOrMore>
```

#####

1.3.16.6.

1: #### oneOrMore##

oneOrMore## ##### oneOrMore##

#1

```
<oneOrMore>  
  <ref name="a" />  
</oneOrMore>  
<oneOrMore>  
  <ref name="b" />  
</oneOrMore>
```

#2

```
<oneOrMore>  
  <ref name="a" />  
  <ref name="b" />  
</oneOrMore>
```

#1####a#1#####b#1#####

(aa...bb...)#####2####a#b#####1##### (abab...)#####

2:

- choice##,interleave## #####
- oneOrMore## #####choice##,interleave## #####

1.3.16.7.

- James Clark and Makoto Murata, ISO/IEC FDIS 19757-2 Document Schema Definition Language (DSDL) -- Part 2: Regular-grammar-based validation -- RELAX NG
- ## #, #oneOrMore## @ ##### RELAX NG ###

1.3.17. zeroOrMore

1.3.17.1. 0#####

- #####(0###)#####

- XML #####
- attribute## ##### (optional##
#####)
- attribute## #####
- grammar #####
- ##### externalRef
#####
- #####

1.3.17.2.

lang:occurs ##

lang

- decode : #####
- encode : #####(encode #####)#

lang:occured ##

lang #####zeroOrMore

- decode : #####
- encode : #####

1.3.17.3.

- optional#oneOrMore#zeroOrMore #####32####
- ref #####define/ref #####
- lang:occurs ####lang:occured #####
- lang:occurs ### 0 #####

1.3.17.4.

- #### ##### interleave##, optional##,
zeroOrMore## #####

1.3.17.5.

1: #### zeroOrMore##

zeroOrMore## ##### zeroOrMore##

#1

```
<zeroOrMore>
<ref name="a" />
```



```
</zeroOrMore>  
<zeroOrMore>  
  <ref name="b" />  
</zeroOrMore>
```

#2

```
<zeroOrMore>  
  <ref name="a" />  
  <ref name="b" />  
</zeroOrMore>
```

#1####a#1#####b#0#####

(aa...bb...)#####2####a#b#####0##### (abab...)#####

2:

- choice##, interleave## #####
- zeroOrMore## #####choice##, interleave##
#####

1.3.17.6.

- James Clark and Makoto Murata, ISO/IEC FDIS 19757-2 Document Schema Definition Language (DSDL) -- Part 2: Regular-grammar-based validation -- RELAX NG
- ## #, #zeroOrMore## @ ##### RELAX NG ###

1.3.18. text

1.3.18.1.

- #####(DTD # #PCDATA ##)#####(DTD # CDATA
###)##### (#####!!) #####
- grammar #####
- ##### externalRef
#####

1.3.18.2.

- #####

1.3.18.3.

- #####

1.3.18.4.

1:

- ##### text## ##### text##
#####(#####)#
- choice## ##### text##### text## #####

1.3.18.5. #####

- James Clark and Makoto Murata, ISO/IEC FDIS 19757-2 Document Schema Definition Language (DSDL) -- Part 2: Regular-grammar-based validation -- RELAX NG
- ## #, #text## @ ##### RELAX NG ###

1.3.19. value ##

1.3.19.1. #####

- ##### value## #####
- element## ##### attribute## #####
- grammar #####

1.3.19.2. ##

type ##

#####

- type ## #####(#####)### datatypeLibrary ##
#####value## #####
- type ## ##### type="token" #(#####)#####
datatypeLibrary ## #####(RELAX NG #####)#

1.3.19.3. ##

value ##### Mapping definition #####

- - lang:value-of - lang:value -
 - - dat:container - dat:value -
 - - dat:container - lang:value-of - ums:value -
- ##### dat:container # dat:byte, dat:bit, rng:element, rng:attribute ##### obsolete
###
- - lang:value-of - dat:container - ums:value -

1.3.19.4. #####

#####

1.3.19.5.

1:

#####

####(###)

```
<element name="##">
  <choice>
    <value>###</value>
    <value>#####</value>
    <value>#####</value>
  </choice>
</element>
```

XML ##(###)###

- <##>#####</##> <!-- ### -->
- <##>#####</##> <!-- ### -->

2:

##

#####(#####)#

####(###)

```
<element name="##">
  <empty/>
  <attribute name="####">
    <text/>
  </attribute>
  <attribute name="##">
    <choice>
      <value>#</value>
      <value>#</value>
    </choice>
  </attribute>
</element>
```

XML ##(###)###

- <## ##="#RheineRiver.jpg" ##="#"/> <!-- ### -->
- <## ##="#ObiRiver.gif" ##="#"/> <!-- ### -->

#####value## ##choice##

#####choice## #####value##

#####

DTD

#####RELAX-NG#####XML#####

1.3.19.6. #####

- James Clark and Makoto Murata, ISO/IEC FDIS 19757-2 Document Schema Definition Language (DSDL) -- Part 2: Regular-grammar-based validation -- RELAX NG
- ## #, #value## @ ##### RELAX NG ###

1.3.20. data ##

1.3.20.1. #####

- #####(#####)# URI, Base64 ##### data##
#####
- #####element##, attribute## ##### data##
#####
- ##### datatypeLibrary ## #####

1.3.20.2. ##

type ##

#####(#####)#####

1.3.20.3. ##

data ##### Mapping definition #####

- - lang:value-of - lang:data -
- - dat:container - dat:data -
- - dat:container - lang:value-of - ums:data -

dat:container # dat:byte, dat:bit, rng:element, rng:attribute ##### obsolete
###

- - lang:value-of - dat:container - ums:data -

#####

- decode ## lang:data
- encode ## dat:data

1.3.20.4. #####

- ##### type ## #####
- param## #####param## #####

1.3.20.5. ###

1:

W3C XML Schema Part 2 #####param##
#####param#####

##, ##, ##, ##URI
#####(#####)#

#####

```
<?xml version="1.0"?>
<element name="#####" xmlns="http://relaxng.org/ns/structure/1.0"
  datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes">
  <element name="##">
    <data type="string"/> <!-- ##### -->
  </element>
  <element name="##">
    <data type="integer"/> <!-- ##### -->
  </element>
  <element name="###">
    <data type="date"/> <!-- #####(ISO 8601##) -->
  </element>
  <element name="###">
    <data type="anyURI"/> <!-- ##### URI -->
  </element>
</element>
```

XML

```
<?xml version="1.0"?>
<#####>
  <##>#####</##>
  <##>18</##>
  <###>1985-05-11</###>
  <###>http://www.sanada.org/</###>
</#####>
```

XML

```
<?xml version="1.0"?>
<#####>
  <##>#####</##>
  <##>#####</##> <!-- ##### -->
  <###>##40#5#1#</###> <!-- ISO 8601 ##### -->
  <###>#####</###> <!-- URI ## -->
</#####>
```

1.3.20.6.

- James Clark and Makoto Murata, ISO/IEC FDIS 19757-2 Document Schema Definition Language (DSDL) -- Part 2: Regular-grammar-based validation -- RELAX NG
- ## #, #data## @ ##### RELAX NG ###

1.3.21. param

1.3.21.1.

- #####
- name ## #####name ## ##### param##
#####(RELAX-NG #####)#
- ## data## #####

1.3.21.2.

name ##

#####(RELAX-NG #####)#

1.3.21.3.

- #####(RELAX-NG
#####)#
- ##### param## #####

1.3.21.4.

1:

data#####, #####

#####

- #####1 # 10 #####
- #####0-150#####

#####

#####

```
<?xml version="1.0"?>
<element name="#####" xmlns="http://relaxng.org/ns/structure/1.0"
  datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes">
  <element name="##">
    <data type="string">
      <param name="minLength">1</param>
      <param name="maxLength">10</param>
    </data>
  </element>
  <element name="##">
    <data type="integer">
```

```
<param name="minInclusive">0</param>
<param name="maxExclusive">150</param>
</data>
</element>
</element>
```

XML

```
<?xml version="1.0"?>
<#####>
<##>#####</##> <!-- ##5## -->
<##>18</##> <!-- ##1-150#### -->
</#####>
```

XML

```
<?xml version="1.0"?>
<#####>
<##>#####</##> <!-- 11##! -->
<##>250</##> <!-- 150###! -->
</#####>
```

1.3.21.5.

- James Clark and Makoto Murata, ISO/IEC FDIS 19757-2 Document Schema Definition Language (DSDL) -- Part 2: Regular-grammar-based validation -- RELAX NG
- ## #, #param## @ ##### RELAX NG ###

1.3.22. except

1.3.22.1.

- #####/##### except## #####
- ##/##### anyName##, nsName## #####
- data## #####except##
#####element##,
attribute## #####

1.3.22.2.

- #####

1.3.22.3.

except## #####nsName##, name##, choice## #####

1.3.22.4.

1:

#####

```
<element name="####">
  <zeroOrMore>
    <element>
      <anyName>
        <except>
          <name>##</name>
          <name>##</name>
        </except>
      </anyName>
      <text/> <!-- ##### -->
    </element>
  </zeroOrMore>
</element>
```

#####

2:

 ##(###)#####choice##

####(###)##

```
<element name="###">
  <choice>
    <zeroOrMore>
      <element>
        <anyName>
          <except>
            <name>###</name> <!-- ##### -->
          </except>
        </anyName>
        <text/>
      </element>
    </zeroOrMore>
    <element name="####"> <!-- ##### -->
      <empty/>
    </element>
  </choice>
</element>
```

XML ##(###)1

```
<###>
  <##>###=#####</##>
  <##>16</##>
  <#####>#####</#####>
</###>
```

XML ##(###)2

```
<###>
  <####/>
</###>
```


XML

```
<###>
  <name>Y. Imada</name>
  <####/> <!-- #### ##### !! -->
</###>
```

3:

#####URI http://www.evil.com/
#####

```
<element name="###">
  <zeroOrMore>
    <element>
      <anyName>
        <except>
          <nsName ns="http://www.evil.com/" />
        </except>
      </anyName>
      <text/> <!-- ##### -->
    </element>
  </zeroOrMore>
</element>
```

4:

#####(### undefined # ##,
none)#####

```
<element name="##">
  <data type="string">
    <except>
      <choice>
        <value>undefined</value>
        <value>###</value>
        <value>none</value>
      </choice>
    </except>
  </data>
</element>
```

#####(### ##, ##, # #####)#####
W3C XML Schema #####

```
<element name="####">
  <data type="string"
  datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes">
    <except>
      <data type="string">
        <param name="pattern">[.\s]*(##|##|#)[.\s]*</param>
      </data>
    </except>
```

```
</data>
</element>
```

1.3.22.5. #####

- James Clark and Makoto Murata, ISO/IEC FDIS 19757-2 Document Schema Definition Language (DSDL) -- Part 2: Regular-grammar-based validation -- RELAX NG
- ## #, #except## @ ##### RELAX NG ##

1.3.23. empty ##

1.3.23.1. #####

- XML#####
- #####XML#####
- #####
- ##### # ##### (XML#####) #####
- grammar #####

1.3.23.2. ##

- #####

1.3.23.3. #####

- #####

1.3.23.4. #####

1: ##### empty##

- ##### empty## ##### empty## #####
- interleave##, group##, oneOrMore##, zeroOrMore##, optional## #####
empty## ##### empty## #####empty##
empty##
- choice## ## empty## #####

2: choice##

- choice## ##### empty## (#####)#####choice## #####
empty## #####
- choice## ##### empty##
<choice>
<value>###</value> <value>##</value> <value>####</value> <empty/> </choice>
#####empty##

```
#####  
choice## #####
```

3: element##

```
element## ##### empty## (#####)##### element##  
#####
```

1.3.23.5. ####

- James Clark and Makoto Murata, ISO/IEC FDIS 19757-2 Document Schema Definition Language (DSDL) -- Part 2: Regular-grammar-based validation -- RELAX NG
- ## #, #empty## @ #### RELAX NG ###

1.3.24. notAllowed ##

1.3.24.1. #####

- #####
- choice##
#####
- notAllowed## #####(choice## ##)#####
- #####
- #####(#####) notAllowed## #####
- grammar #####

1.3.24.2. ##

- #####

1.3.24.3. ####

- #####

1.3.24.4. #####

1: choice## ### notAllowed##

- choice## ##### notAllowed## #####notAllowed##
#####
- #####choice## ### notAllowed## ##### notAllowed## #####
(notAllowed## ##### interleave## ##)##### choice## #####
notAllowed## #####

2:

- ##### notAllowed## ##### notAllowed## #####
- notAllowed## ##### notAllowed##
#####notAllowed## ##### attribute##, group##,
interleave##, oneOrMore## #####
- zeroOrMore##, optional## ##### notAllowed##
(notAllowed## #####)#
- except## ##### notAllowed## ##### except## #####
- notAllowed## ##### element## #####
(#####)#

1.3.24.5.

- James Clark and Makoto Murata, ISO/IEC FDIS 19757-2 Document Schema Definition Language (DSDL) -- Part 2: Regular-grammar-based validation -- RELAX NG
- ## #, #notAllowed## @ ##### RELAX NG ##

1.3.25. DATA

1.3.25.1. Text Binary

bit ##

- #####
- #####
- ##
- length ##
- #####
- encode ##
- #####
- ##### txt, ##### signed, unsigned, ieee754single,ieee754double #####
- ##### encode ##### txt #####
- #####
- ##### (byte##, bit##) #####
- interleave##, optional##, zeroOrMore## #####
- #####value-of #####

umsCodeGenerator

###

1: length

#####10 bit #####

```
<txt:bit length="10" encode="signed">
  <data type="int"/>
</txt:bit>
```

2: length

#####type # int #####32 bit #####

```
<txt:bit encode="signed">
  <data type="int"/>
</txt:bit>
```

3: bit

bit #####length ##### length ##### bit
signed type ##### 10 bit#unsignedInt type ##### 6 bit

```
<txt:bit length="16">
  <txt:bit length="10" encode="signed">
    <data type="int"/>
  </txt:bit>
  <txt:bit encode="unsigned">
    <data type="unsignedInt"/>
  </txt:bit>
</txt:byte>
```

byte ##

#####

- #####

##

length ##

#####

```

encode ##
#####
• ##### txt, ##### signed, unsigned, ieee754single,ieee754double #####
• ##### encode ##### txt #####

####
• ##### (byte##, bit##) #####
• interleave##, optional##, zeroOrMore## #####
• #####value-of #####

line ##

1#####
• #####

##

encode ##
#####
• ##### txt, ##### signed, unsigned, ieee754single,ieee754double #####
• ##### encode ##### txt #####

##
• defineMapping #####

####
• ##### (byte##, bit##) #####

###

# 1: line #####
#####1#####

```

```

<txt:line>
  <txt:byte encode="txt">
    <lang:value-of select="iData">
      <data type="int"/>
    </lang:value-of>
  </txt:byte>
</txt:line>

```

2: line

1 #####line ##### value

```
<txt:byte encode="txt">
  <lang:value-of select="iData">
    <data type="int"/>
  </lang:value-of>
</txt:byte>
<txt:byte encode="txt">
  <txt:value type="string">
</txt:value>
</txt:byte>
```

list ##

#####

- #####list## #####
- #####, ##, #####
- XML ##### element##, attribute##, define## #####
- grammar #####
- ##### externalRef #####

##

separator ##

#####

- decode : #####1#####
- encode : ### token #####

delimiter ##

#####

- decode : #####
- encode : ### token #####

####

separator ##### "##\t##\n##\r"#####

```
##
#####
#A,,B##### 0 #####
```

- ```
####
```
- value##, data## #####text##  
#####
  - XML ## element##, attribute## #####
  - #####(#####, #####)#####interleave##,  
choice## #####

```
###
```

```
1:
```

```
#####
XML ##(###)#####
```

**#####**

```
<element name="#">
 <attribute name="##"><text/></attribute>
 <list>
 <oneOrMore>
 <choice>
 <value>#</value>
 <value>#</value>
 <value>#</value>
 <value>#</value>
 </choice>
 </oneOrMore>
 </list>
</element>
```

**### XML #####**

```
<# ##="#####">
#
</#>
```

**##### XML #####**

```
<# ##="#####">
<!-- "#" ##### -->
</#>
```

```
2:
```



Cascading Style Sheets ### #####(#, #, #, ###)##### XML  
###(#####)#####list## ##### RGB.datatype  
##### (#RRGGBB #####) ##### (#, #, ##)#####

#####

```
<element name="##">
 <empty/>
 <attribute name="#">
 <list>
 <oneOrMore>
 <choice>
 <ref name="RGB.datatype"/>
 <value>#</value>
 <value>#</value>
 <value>##</value>
 </choice>
 </oneOrMore>
 </list>
 </attribute>
</element>
```

### XML #####

- <## #=" #222255 #ab7766 #00a3b1 #ff0022 " />
- <## #=" ## #ab7766 # " />

#####

####

- James Clark and Makoto Murata, ISO/IEC FDIS 19757-2 Document Schema Definition Language (DSDL) -- Part 2: Regular-grammar-based validation -- RELAX NG
- ## #, #list## @ ##### RELAX NG ###

### 1.3.25.2. XML

element ##

#####

- #####
- XML#####
- #####

##

name ##

- ##### Qualified name #####

```

• name ## #####name## #####
• #####anyName## #####except##
 #####

ns ##

URI

xmlns:xxxx ##

grammar## #####xmlns:xxxx ## #####

xmlns ##

element## #####
(http://relaxng.org/ns/structure/1.0) #####

####
• element## (## element## ##### choice## ##)
 #####
• #####element## #####choice##, zeroOrMore##
 #####
• element## (#####element##### choice## ##)
 #####element## #####
• (#####)##### element##
 #####
• ref #####

###

1:

URI http://lena.com #### p

<?xml version="1.0"?>
<element name="p" ns="http:lena.com"
xmlns="http://relaxng.org/ns/structure/1.0">
 <text/>
</element>

2:

p ## #### p ##### a1, a2 ## a1, a2 #####

<element name="p">

```

## umsCodeGenerator

```
<element name="a1"><text/></element>
<element name="a2"><text/></element>
</element>
```

# 3:

book ##### book ##### book #####

```
<element>
 <anyName>
 <except><name>book</name></except>
 </anyName>
 <element name="book"><text/></element>
</element>
```

####

- James Clark and Makoto Murata, ISO/IEC FDIS 19757-2 Document Schema Definition Language (DSDL) -- Part 2: Regular-grammar-based validation -- RELAX NG
- ## #, #element## @ #### RELAX NG ###

**attribute ##**

#####

- lang # space #####
- ### attribute## #####element## #####  
#####attribute##  
#####
- element## #####(DTD ### #REQUIRED #####)#####  
(DTD ### #IMPLIED) #####optional## ## attribute## #####
- oneOrMore##, zeroOrMore##  
#####optional## #####

##

**name ##**

- #####element## ##### name## ##### anyName##  
#####(qualified  
name)#####

**ns,xmlns:xxxx ##**

- grammar##, element## #####ns ## # xmlns:xxxx ##  
#####
- ##### xmlns:xxxx ## ##attribute##  
#####

- ```
####
```
- #####
 - text## #####value##, data## # choice## #####
 - element##, attribute## #####
 - #####ref## #####
 - ref #####

###

1:

item ##### lang ####xlink:href ##### lang ##### xlink:href
#####

```
<element name="p">
  <optional>
    <attribute name="lang">

      <text/>
    </attribute>
  </optional>
  <attribute name="xlink:href" xmlns:xlink="http://www.w3.org/1999/xlink">

    <text/>
  </attribute>
  <text/> <!-- p##### -->
</element>
```

2:

color ##### red, blue, yellow #####

```
<attribute name="color">
  <choice>
    <value>red</value>
    <value>blue</value>
    <value>yellow</value>
  </choice>
</attribute>
```

####

- James Clark and Makoto Murata, ISO/IEC FDIS 19757-2 Document Schema Definition Language (DSDL) -- Part 2: Regular-grammar-based validation -- RELAX NG
- ## #, #attribute## @ ##### RELAX NG ###

mixed ##

umsCodeGenerator

```
#####(###)#####  
• ##### mixed##  
#####  
• grammar #####  
• ##### externalRef  
#####  
  
##  
• #####  
  
####  
• #####  
• #####  
• ##### value##, data## #####text##  
#####  
• element##, attribute## #####  
• #####(#####, #####)#####interleave##,  
choice## #####  
  
###  
  
# 1:  
##### ##, ##,  
#####(###)#####  
  
####  
  
<element name="##">  
  <mixed>  
    <element name="##"><text/></element>  
    <element name="##"><text/></element>  
    <element name="##"><text/></element>  
  </mixed>  
</element>  
#####(###)#  
  
<##>  
###<##>####</##>#<##>###</##>##  
<##>202###</##>#####  
</##>  
  
# 2:  
interleave##, zeroOrMore## #####a### b
```

#####mixed##
#####

mixed #####

```
<element name="element">
  <mixed>
    <interleave>
      <zeroOrMore>
        <element name="a"><text/></element>
      </zeroOrMore>
      <zeroOrMore>
        <element name="b"><text/></element>
      </zeroOrMore>
    </interleave>
  </mixed>
</element>
```

choice #####

```
<element name="element">
  <zeroOrMore>
    <choice>
      <element name="a"><text/></element>
      <element name="b"><text/></element>
    </choice>
  </zeroOrMore>
</element>
```

#####choice## #####choice##
#####

#####a, b#####a ###0#### b #####
mixed## # interleave## #####

```
<element name="element">
  <mixed>
    <interleave>
      <zeroOrMore>
        <element name="a"><text/></element>
      </zeroOrMore>
      <element name="b"><text/></element>
    </interleave>
  </mixed>
</element>
```

#####

1:

empty##, text## #### attribute## ##### mixed## ##### text## #####

####

- James Clark and Makoto Murata, ISO/IEC FDIS 19757-2 Document Schema Definition Language (DSDL) -- Part 2: Regular-grammar-based validation -- RELAX NG
- ## #, #mixed## @ #### RELAX NG ##

name ##

##, #####

- #####element##, attribute## # name ## #####name##
#####
- name## ##, #####

#####

- name## #####(qname)#####
- #####
- #####choice## #####
- name## #####element##, attribute## # name ## #####

##

- #####

###

1: name## # name

#####(###)#####

#1

```
<element name="#:##">  
<text/>  
</element>
```

#2

```
<element>  
<name>#:##</name>  
<text/>  
</element>
```

2:

#####(##,
#####)#####

####

```
<?xml version="1.0"?>
<element xmlns="http://relaxng.org/ns/structure/1.0">
  <choice>
    <name>##</name>
    <name>##</name>
  </choice>
  <element name="##"><text/></element>
  <element name="##"><text/></element>
</element>
```

XML ###1

```
<?xml version="1.0"?>
<##>
  <##>#####</##>
  <##>#####</##>
</##>
```

XML ###2

```
<?xml version="1.0"?>
<##>
  <##>#####</##>
  <##>#####3-2-1</##>
</##>
```

#####

ns ##

#####

#1

```
<element name="###" ns="http://www.sanada.org/">
  <attribute name="####">
    <text/>
  </attribute>
  <empty/>
</element>
```

#2

```
<element name="###" ns="http://www.sanada.org/">
  <attribute>
    <name>####</name>
    <text/>
  </attribute>
  <empty/>
</element>
```

####xmlns:xxxx ## #####:####
#####

#1

```
<element name="##:###" xmlns:##="http://www.sanada.org/">
```



```
<attribute name="##:####">
  <text/>
</attribute>
<empty/>
</element>
```

#2

```
<element name="##:####" xmlns:##="http://www.sanada.org/">
  <attribute>
    <name>##:####</name>
    <text/>
  </attribute>
  <empty/>
</element>
```

####

- James Clark and Makoto Murata, ISO/IEC FDIS 19757-2 Document Schema Definition Language (DSDL) -- Part 2: Regular-grammar-based validation -- RELAX NG
- ## #, #name## @ #### RELAX NG ###

anyName ##

#####

- #####(#####)#####
anyName## #####
- name ## ##### element##, attribute## #####anyName##
element##, attribute## ### name ##

##

- #####

####

- anyName## #####
- except## #####except## #####

###

1:

```
### ######(#####)#####  
#####
```

####

```
<element name="####"> <interleave> <!-- ##### -->  
<zeroOrMore> <element> <!-- ##### --> <anyName> <except>
```

```

<name>##</name> </except> </anyName> <text/> </element> </zeroOrMore>
<element name="##"><text/></element> </interleave> </element>
### XML ##(###)
<####> <##>####</##> <!-- ##### --> <##>####</##> <##>#####</##>
</####>
#### XML ##(###)
<####> <!-- ##### --> <##>####</##> <##>##</##> </####>

```

####

- James Clark and Makoto Murata, ISO/IEC FDIS 19757-2 Document Schema Definition Language (DSDL) -- Part 2: Regular-grammar-based validation -- RELAX NG
- ## #, #anyName## @ #### RELAX NG ###

nsName ##

#####

- #####, ##### nsName## #####
- #####anyName## ##### anyName##
#####nsName## #####
- (#####) nsName## ##### element##, attribute## ## name ##
#####

##

ns ##

- ##### ns ## #####xmlns ##
#####(#####)#
- ns ## ##### ns ## #####

####

anyName## ####except## #####except## # name##
#####, #####

###

1:

http://www.garden.co.jp/

```

<element name="###">
<zeroOrMore>

```

umsCodeGenerator

```
<attribute>
  <nsName ns="http://www.garden.co.jp/" />
  <text/>
</attribute>
</zeroOrMore>

...
</element>
```

XML ##(###)#####

```
<### #:##="#####" #:##="10m^2"
  xmlns:##="http://www.garden.co.jp/">
...
</###>
```

ns ## ##### nsName##

```
<element name="###">
  <zeroOrMore>
  <attribute>
    <nsName ns="http://www.garden.co.jp/" />
    <nsName ns="http://www.address.co.jp/" />
    <text/>
  </attribute>
  </zeroOrMore>

...
</element>
```

2:

except## #####1 #####

```
<element name="###">
  <zeroOrMore>
  <attribute>
    <nsName ns="http://www.garden.co.jp/">
    <except>
      <name>##</name>
    </except>
  </nsName>
  <text/>
  </attribute>
  </zeroOrMore>

...
```

```
</element>
```

```
##### #:## #####
```

```
<### #:##="#####" #:##="10m^2"
  xmlns:##="http://www.garden.co.jp/">
  ...
```

```
# 3: #####
```

```
except## ### nsName## #####
http://bad.com/ #####
```

```
<element name="#">
  <zeroOrMore>
    <element>
      <anyName>
        <except><nsName ns="http://bad.com/" /></except>
      </anyName>
    </element>
  </zeroOrMore>
</element>
```

```
####
```

- James Clark and Makoto Murata, ISO/IEC FDIS 19757-2 Document Schema Definition Language (DSDL) -- Part 2: Regular-grammar-based validation -- RELAX NG
- ## #, #nsName## @ #### RELAX NG ###

1.3.26. LANG

1.3.26.1. function

```
#####
```

- #####
- #####defineMapping #####
- #####defineMapping #####

```
##
```

```
name ##
```

```
#####
```

umsCodeGenerator

##

- defineFunctions #####

####

- arg ###return ###exception ###java #####defineMapping ###defineVariables
#####

###

1:

```
<lang:function name="decodeData">
  <lang:arg type="char*" name="ums__buffer" direction="in"/>
  <lang:arg type="int" name="ums__bitlen" direction="in"/>
  <lang:return type="void"/>

  <defineVariables>
    <lang:var type="int" name="iData"/>
  </defineVariables>
  <defineMapping direction="decode"
xmlns:txt="http://ums.isas.jaxa.jp/0.4/dat">
    <txt:list encode="txt">
      <txt:data type="token"/>
      <lang:value-of select="iData">
        <data type="int"/>
      </lang:value-of>
      <lang:callFunction expr="setIntValue( iData );"/>
    </txt:list>
  </defineMapping>
</java:function>
```

1:

```
<lang:function name="setIntValue">
  <lang:arg type="int" name="iData" direction="in"/>
  <lang:return type="void"/>
</lang:function>
```

1.3.26.2. arg

#####

- #####
- #####

```
##

  type ##
#####

  class ##
#####class #####java #####

  name ##
#####

  direction ##
#####in#out#inout #####
  • in : #####
  • out : #####
  • inout : #####

##
  • function #####

####
  • #####

###

# 1:
```

```
<lang:function name="setValue">
  <lang:arg type="int" name="iData" direction="in"/>
  <lang:arg type="unsigned int" name="uiData" direction="in"/>
  <lang:return type="void"/>
</lang:function>
```

1.3.26.3. return ##

```
#####
  • #####
```

umsCodeGenerator

##

type ##

#####

##

- function #####
- function #####

####

- #####

###

1:

```
<java:function scope="abstract" name="setIntValue">
  <java:arg class="int" name="iData" direction="in"/>
  <ava:return type="void"/>
</java:function>
```

1.3.26.4. var

#####

- #####

##

type ##

#####

class ##

#####class #####java #####

name ##

#####

##

- defineVariables ###struct ##### clng #####

####

- #####

###

1:

```
<defineVariables>
  <lang:var type="int" name="iData"/>
</defineVariables>
```

1.3.26.5. array

#####

- #####

##

size ##

#####

##

- defineVariables ###struct ##### clng #####

####

- var #####

###

1:

```
<defineVariables>
  <lang:array size="4096">
    <lang:var type="int" name="iData"/>
  </lang:array>
</defineVariables>
```

1.3.26.6. value-of

umsCodeGenerator

#####

##

select ##

(C #####)

##

Mapping definition #####

- - lang:value-of - lang:data -
- - lang:value-of - lang:value -
- - dat:container - lang:value-of - ums:data -
- - dat:container - lang:value-of - ums:value -

dat:container # dat:byte, dat:bit, rng:element, rng:attribute ##### obsolete
###

- - lang:value-of - dat:container - ums:data -
- - lang:value-of - dat:container - ums:value -

####

#####

###

1: mapping ## (proposal for v0.4)

XML-XML syntax ## mapping defition

```
<bin:byte length="1">
  <lang:value-of select="var">
    <data type="int"/>
  </lang:value-of>
</bin:byte>
```

- ### encode # decode # ##### decode ##### var ##### encode
var

XML-language syntax ##### ##decode

```
<bin:byte length="1">
```

```
var = <data type="int"/>;
</bin:byte>
```

encode

```
<bin:byte length="1">
  <data type="int"/> = var;
</bin:byte>
```

XML-language syntax ##### ##### ###mapping definition
XSL ##### XML-XML syntax

##

#####(-KM-)#

1.3.26.7. callFunction

#####

- #####

##

expr ##

#####

##

- defineMapping #####

####

- #####

###

1:

```
<lang:callFunction expr="setValue( iData );"/>
```

1.3.26.8. synchronized

#####

umsCodeGenerator

- #####
- #####"##"#####
#####synchronized #####"##"##1#####

##

- #####

##

- defineMapping #####

####

- callFunction ###value-of #####

###

1:

#####nextData() #####"##"#####

```
<lang:synchronized>  
  <lang:callFunction expr="nextData();" />  
</lang:synchronized>
```

1.3.26.9. C

struct ##

C#####

- #####

##

name ##

#####

##

- start #####

####

- ### array ###var #####

###

1:

```
<clng:struct name="Record">
  <clng:array size="4096">
    <clng:var type="char" name="strData"/>
  </clng:array>
  <clng:var type="int32_t" name="iData"/>
  <clng:var type="uint32_t" name="uiData"/>
</clng:struct>
```

1.3.26.10. Java

class ##

Java #####

- #####
- class #####inner class #####
- grammar #####

##

abstract ##

#####

name ##

#####

extends ##

#####

##

- #####

####

- implements ###defineFunctions ###defineVariables #####

###

umsCodeGenerator

1:

```
<ava:class scope="abstract" name="Sample"
xmlns:java="http://ums.isas.jaxa.jp/0.4/java">
  <defineFunctions>
    <java:function scope="abstract" name="setStringValue">
      ...
    </java:function>
    <java:function scope="abstract" name="setIntValue">
      ...
    </java:function>
  </defineFunctions>
  <defineFunctions>
    <java:function name="decodeData">
      ...
      <defineVariables>
        ...
      </defineVariables>
      <defineMapping direction="decode"
xmlns:txt="http://ums.isas.jaxa.jp/0.4/dat">
        ...
      </defineMapping>
    </java:function>
    <java:function name="encodeData">
      ...
      <defineVariables>
        ...
      </defineVariables>
      <defineMapping direction="encode"
xmlns:txt="http://ums.isas.jaxa.jp/0.4/dat">
        ...
      </defineMapping>
    </java:function>
  </defineFunctions>
</java:class>
```

implements ##

interface #####

- ##### interface #####
- intervace #####

##

name ##

interface #####

 • class #####

 • #####

###

1:

```
<java:class name="Sample" extends="Data"
xmlns:java="http://ums.isas.jaxa.jp/0.4/java">
  <java:implements name="DataInterface"/>
  <java:implements name="DataInterface2"/>
  ...
</java:class>
```

exception ##

 • #####

##

type ##

#####

 • function #####
 • function #####

 • #####

###

1:

```
<java:function scope="abstract" name="setIntValue">
  <java:arg class="int" name="iData" direction="in"/>
  <ava:return type="void"/>
  <ava:exception type="UMSEException"/>
```

umsCodeGenerator

```
</java:function>
```

2. umsCodeGenerator

2.1. umsCodeGenerator

[English](#)

2.1.1. umsCodeGenerator

```
umsCodeGenerator ##mappingSchema ##### mapping definition ##### UMS
##### encode#decode#####

##### mappingSchema ##### UMS group #####
mappingSchema #####

umsCodeGenerator #####
```

Note:

```
### umsCodeGenerator # tableTools ##### umsCodeGenerator # version 0.4
##### UMS group #####'Brief History' #####
```

2.1.2.

```
tableTools#####
#####", "##### CSV#####
```

```
csvdata1,11,1.2345
```

```
###XML#####
```

```
<Record>
  <strData>xmldata1</strData>
  <intData>11</intData>
  <doubleData>1.2345</doubleData>
</Record>
```

```
#####
tableTools#####
#####
#####
```

2.1.3.

#####

	Java	C	C++	Perl
Text	#	#	--	--
Binary	#	#	--	--
XML	--	--	--	--

#####

#####decode#####

encode#####

2.1.4. #####

Mapping definition ## ##### (####) #####

Text, Binary ##

- [byte](#)
- [bit](#)

XML ##

- #
- ##
- [list](#)

2.1.5. #####

#####

Java

class / function / arg / return / exception / array / var

C

package / function / arg / return / struct / array / var

C++

#

Perl

#

#####

##

- [value-of](#)
- callFunction

2.1.6. #####

umsCodeGenerator

#####

- data
- value (#####)

2.1.7. encodingLibrary

umsCodeGenerator ##### encode ### [W3C datatype](#)
 (http://www.w3.org/TR/xmlschema-2/#built-in-datatypes) # #####
 (encodingLibrary) #####

encode	W3C datatype	Language			
		Java	C	C++	Perl
txt	string	String	char *	--	--
	token	String	char *	--	--
	byte	byte	int8_t	--	--
	short	short	int16_t	--	--
	int	int	int32_t	--	--
	long	long	int64_t	--	--
	unsignedByte	short	uint32_t	--	--
	unsignedShort	int	uint32_t	--	--
	unsignedInt	long	uint32_t	--	--
	unsignedLong	BigInteger	uint32_t	--	--
signed	byte	byte	int8_t	--	--
	short	short	int16_t	--	--
	int	int	int32_t	--	--
	long	long	int64_t	--	--
unsigned	unsignedByte	short	uint32_t	--	--
	unsignedShort	int	uint32_t	--	--
	unsignedInt	long	uint32_t	--	--
	unsignedLong	BigInteger	uint32_t	--	--

ieee754double	float	float	float	--	--
	double	double	double	--	--

2.1.8. #####

#####

- JAVA: JAVA2 1.4
- C#: ISO9899 (C99)

2.2. umsCodeGenerator #####

[English](#)

2.2.1. #####

umsCodeGenerator #####

- [Java 2 Platform, Standard Edition \(J2SE\)](http://java.sun.com/) ##### 1.4 ##
Java ##### JAVA_HOME ##### PATH # \$JAVA_HOME/bin #####
###umsCodeGenerator #####
- [Apache Ant](http://ant.apache.org/) (http://ant.apache.org/) ##### 1.6.5
Java ##### build ####
- [msv](https://msv.dev.java.net/) (https://msv.dev.java.net/)
RELAX NG#RELAX Namespace#RELAX Core#TREX#XML DTDs#XML Schema
Part 1#####
- [jing](http://www.thaiopensource.com/relaxng/jing.html) (http://www.thaiopensource.com/relaxng/jing.html)
RELAX NG #####
- [trang](http://thaiopensource.com/relaxng/trang.html) (http://thaiopensource.com/relaxng/trang.html)
#####
- [Xalan-Java](http://xml.apache.org/xalan-j/) (http://xml.apache.org/xalan-j/) Apache XML Project ##### XSLT
#####

2.2.2. #####

umsCodeGenerator ##### version 0.4(2009032501) ### #####

- [umsCodeGenerator-2009032501.tar.gz](#)

2.2.3. #####

#####

umsCodeGenerator/yyyymddvv #####

umsCodeGenerator

```
$ tar zxvf umsCodeGenerator-yyyyymmddvv.tar.gz
umsCodeGenerator/yyyyymmddvv ### #####setup.sh # setup_cygwin.sh#
#####
#####
$ source setup.sh
#####C Compiler #####
$ export CC="gcc -std=c99 -g"
#####umsCodeGenerator # build #####
$ cd $TABLETOOLS_HOME
$ ant
#####
##### "BUILD
SUCCESSFUL" #####
$ ant test
```

2.2.4.

Copyright 2005 Universal Mapping Schema group and ISAS/JAXA
Apache License Version 2.0#####

<http://www.apache.org/licenses/LICENSE-2.0>
(<http://www.opensource.jp/licenses/apache2.0.html>)

#####

2.3. umsCodeGenerator

2.3.1. umsCodeGenerator

1. umsCodeGenerator #####
 2. #### UMS_THREAD ##"true" ### "false" #####
"true" ##### "true" ## "false"
#####
- ```
$ export UMS_THREAD="true"
###
```

```

$ export UMS_THREAD="false"
3. C #####
$ txt2clng ums_file -x|-l [output_directory] [-d]
 Java #####
$ txt2java ums_file -x|-l [output_directory] [-d]

```

2.3.2. #####

| #####            | ##                             |
|------------------|--------------------------------|
| ums_file         | ## UMS #####                   |
| -x               | XML syntax # UMS #####         |
| -l               | Language syntax # UMS<br>##### |
| output_directory | #####<br>#####                 |
| -d               | #####                          |

2.4. #####

2.4.1. #####

```

umsCodeGenerator ##### CSV
#####
1. #####umsCodeGenerator #####
2. ##### mapping definition #####
3. ##### mapping definition #####
4. ##### umsCodeGenerator #####
2###3##### umsCodeGenerator #####
#####
#####umsCodeGenerator #####
umsCodeGenerator/yyyyymmddvv/sample/tutorial
#####Java ## XML syntax#Language syntax#C ## XML syntax#Language syntax
#####

```

### 2.4.2. 1. #####

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

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

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

### 2.4.3. 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#####

### 2.4.3.1. 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 #####
mappingSchema #####

```

**2.4.3.2. 2.2 #####**

```

#####
#####"decode"#####"encode"#####
#####
#####(#####)##umsCodeGenerator ##### (mappingSchema
#####)#

```

| ##   | ####  | #####                                                                                                                                                                                                                                                 |
|------|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| C##  | ##### | <pre> /** * ##### * * @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 ) </pre> |
|      | ##### | <pre> /** * ##### * * @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 ) </pre>         |
| Java | ##### | <pre> /** * ##### * * @param ums__buffer ##### * @param ums__bitlen #####(bit) * ##### </pre>                                                                                                                                                         |

|  |       |                                                                                                                                                                                                                     |
|--|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  |       | <pre>* @return #####(bit) * @exception UMSEException ##### */ int decode( byte[] ums__buffer, int ums__bitlen ) throws UMSEException</pre>                                                                          |
|  | ##### | <pre>/** * ##### * * @param ums__buffer ##### * @param ums__bitlen #####(bit) * @return #####(bit) * @exception UMSEException ##### */ int encode( byte[] ums__buffer, int ums__bitlen ) throws UMSEException</pre> |

#####

```
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="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>

```

#### 2.4.3.3. 2.3 #####

##### mappingSchema ##### XML #####

A,100,1.1

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

```

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

```

#### 2.4.3.4. 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"

```



## umsCodeGenerator

```
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="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>
```

### 2.4.3.5.2.5 #####

#####

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

 int decode(byte[] ums__buffer, int ums__bitlen) throws UMSException {
 :
 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>

 <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>

```

## umsCodeGenerator

```
<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  
#####

### 2.4.4. 3. #####

##### main #####

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

#### 2.4.4.1. 3.1 #####

#####

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

#####

##	#####	#####	####
C##	ums.h	tableTools_init()	tableTools_end()
Java	jp.jaxa.isas.ums.runtime.	*UMSLibrary.tableTools_	ritUMSLibrary.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);
 }
 }
 }
}
```

## umsCodeGenerator

```

 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();
}
}

```

### 2.4.4.2. 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 {

```

## umsCodeGenerator

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

UMSLibrary.tableTools_end();
}
```

```
main #####"Main.java"
```

### 2.4.5. 4. umsCodeGenerator ###

```
mapping definition ## ##### umsCodeGenerator #####
#####"Sample.java" #####
```

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

#### Warning:

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

### 2.4.6. 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>
```

## 2.5. Tips

### 2.5.1. Tips

#### 2.5.1.1. 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
#####
```

#### 2.5.1.2. 2. ##### value ##

```
#####
```

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

```
#####
```

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

#### 2.5.1.3. 3. #####

```
#####
```

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

```
#####
```

```
+ Sample
 + c1ng
 - SampleMain.c
 - SampleTest.c
 - Makefile
 - schemas.xml
 - Sample.ums
```



```
+ java
- SampleMain.java
- SampleTest.java
- build.xml
- schemas.xml
- Sample.ums

+ data
```

#####

#### 2.5.1.4. 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
UMSEException {
 <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:list>
 </dat:byte>
 </defineMapping>
}

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

 UMSELibrary.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>

```

## umsCodeGenerator

```
}
/* ##### */
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>
```

### 2.5.1.5. 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>

```

#### 2.5.1.6. 6. define/ref #####

```
"resolveRef.xml" ##### define/ref ##### XML ##### XSLT #####
```

```
[#####]
src/xslt/raw/resolveRef.xml
```

## 2.6. umsCodeGenerator 0.4

### 2.6.1. Major Versions

#### umsCodeGenerator version 0.4

Capability of

- choice
- repeat
- optional
- zeroOrMore
- oneOrMore.

has been recovered from this version with new processing algorithms

- Method 3: - One pass processing which supports XML, interleave (recurse of self is prohibited), recursive ref-define within finite depth. Sample is shown [here](#).

Supported **Data Structure** and **Processing Language** is the same with tableTools version 0.4.

#### tableTools version 0.4

This version can only handle simple table type structure. Both decoding and encoding of sequential data are supported. On the other hand, following capability was tentatively descope

- choice
- repeat
- optional
- zeroOrMore
- oneOrMore.
- Supported **Data Structure**
  - Sequential Data in Text and Binary format - <http://ums.isas.jaxa.jp/0.4/dat>
- Supported **Processing Language**
  - C - <http://ums.isas.jaxa.jp/0.4/clng>
  - JAVA - <http://ums.isas.jaxa.jp/0.4/java>

#### umsCodeGenerator version 0.3

Most of processing in 'umsCompiler' has been moved into the XSL stylesheet of 'umsCodeGenerator' version 0.3. Since XSLT has limited capability in handling of strings, small portion of processing is still written in JAVA. This version of umsCodeGenerator supports generation of decoding library of serial data. C, JAVA and PERL version was available. Only Method 2 type processing was

supported in this version.

- Supported **Data Structure**
  - Sequential Data in Text and Binary format - <http://ums.isas.jaxa.jp/0.3/dat>
- Supported **Processing Language**
  - Perl - <http://ums.isas.jaxa.jp/0.3/perl>
  - C - <http://ums.isas.jaxa.jp/0.3/clng>
  - JAVA - <http://ums.isas.jaxa.jp/0.3/java>

**umsCompiler version < 0.2**

Oldest version of this program written in JAVA language. This version supported two decoding algorithms

- Method 1 (only for input of text data): Converts mapping definition into a regular expression and use regular expression library.
- Method 2: Two pass processing for both text and binary data.

**2.6.2. Minor Releases**

**2006#11#22#:20061122 # #####**

- #####
- Xalan#MSV#Jing#Trang#####
- W3C datatype # hexBinary #####

**2006#10#20#:20061020 # #####**

- value #####
- #####

**2006#10#06#:20061006 # #####**

- umsCodeGenerator #####
- #####

**2006#09#13#:20060913 # #####**

- ##### UMS\_THREAD #####
- umsCodeGenerator #####

**2006#08#16#:20060816 # #####**

- ##### (mapping definition file) #####

**2006#07#28#:20060728 # #####**

- 20060714 # UMS #####
- Java ## package #####

**2006#07#14#:20060714 # #####**

- 20060419 # UMS #####
- group, interleave, choice, optional, zeroOrMore, oneOrMore, empty #####  
#dat#####

**2006#06#23#:20060623 # #####**

- pre #####umsCodeGenerator ##### build #####

:

**2005#10#25#:** [umsCodeGenerator-2005102501.tar.gz](#)  
#####  
#####  
#####(-KM-)

**2005#9#19#:**  
#####( see <http://ums.isas.jaxa.jp/> )# umsCodeGenerator  
##### line ##### lines #####(-KM-)

**2005#9#17#:** [umsCodeGenerator-2005091701.tar.gz](#)  
#####  
##### encodingLibrary #####(-KM-)

**2005#8#3#:** [umsCodeGenerator-2005080301.tar.gz](#)  
tableTools # umsCodeGenerator ##### file #####mapping definition  
#####(-KM-)

**2005#8#1#:**  
tableTools/umsCodeGenerator ### line ###param ##### line  
#####  
#####umsCodeGenerator # optional, choice,  
zeroOrMore, oneOrMore # #####  
##### CC  
#####(-KM-)

**2005#7#29#:** [tableTools-2005072953.tar.gz](#)  
[umsCodeGenerator-2005072953.tar.gz](#)

- #####
- ##### CC ###
- line #####
- optional ##### (umsCodeGenerator ##)

**2005#7#23#:** [tableTools-2005072305.tar.gz](#)  
[umsCodeGenerator-2005072305.tar.gz](#)  
### syntax (XML-XML syntax) ##### XML-language syntax #####  
##### oneOrMore, zeroOrMore, choice  
#####(umsCodeGenerator)# (-KM-)

**2005#7#2#:** [tableTools-2005070253.tar.gz](#)

- XSL #####
- pre XSL ##### XSL #####
- post xsl #####

**2005#6#26#:** [tableTools-2005062653.tar.gz](#)  
[umsCodeGenerator-2005062653.tar.gz](#)

- #####
- list ##### separator ###delimiter ## #####
- umsCodeGenerator (#####) # tableTools (#####) #####  
##### C #####

**2005#6#24#:** [tableTools-2005062451.tar.gz](#)

- JAVA ##### <list/> ##### CSV ##### separator ##### ##separator  
###delimiter ##### mapping ### re-write #####

- ##### RELAX NG #####
- #####
- 2005#6#21#:**  
##### encodingLibrary ##### (#####)#
- 2005#4#25#:** [tableTools-2005042504.tar.gz](#)  
C ##### <list/> ##### encodingLibrary (RELAX NG # datatypeLibrary #####) # tableTools #####tableTools #####  
encodingLibrary #####
- ##### datatypeLibrary #####
- (C ###) W3C\_typeConversionLibrary #####int ## #####txt #####  
unsignedInt ## #####
- 2005#4#18#:**  
tableTools version 0.4 ##### mappingSchema ### (version 0.4) ##  
#####  
#####(-KM-)
- 2005#4#16#:** [tableTools-20050416.tar.gz](#)
- (C ###) typeConversionLibrary ##### <data/> ###<value/> ### datatype  
#####
- <value/> - string, token ## encode ### ##### \ #####
- <defineMapping/> - #####
- 2005#4#15#:** [tableTools-2005041501.tar.gz](#)
- 2005#4#13#:** [tableTools-2005041301.tar.gz](#)
- [2005#03#####](#)

### 2.6.3. #####

- <value-of>### select ##### a.b.c  
##### <value-of  
select="(a).a"/>#####
- C### typedef #####
- #####Java #####  
C#####
- Java ## typeConversionLibrary ##### encode ##### "US-ASCII" #####
- dataTypeLibrary # URI # type #####

### 2.6.4. #####

See [Bugzilla](#)

([http://eda.plain.isas.jaxa.jp/bugzilla/buglist.cgi?product=UMS&bug\\_status=NEW&bug\\_status=ASSIGNED](http://eda.plain.isas.jaxa.jp/bugzilla/buglist.cgi?product=UMS&bug_status=NEW&bug_status=ASSIGNED))



### 3. Developer's Inforamtion

#### 3.1. umsCodeGenerator

##### 3.1.1. #####

##### 3.1.1.1. #####

```
archive #####testAndTar.sh ##### tar/gz
ums ##### Forrest
```

```
isas/archive
isas/umsCodeGenerator/yyyymddvv
isas/ums
```

**Note:**

```
umsCodeGenerator/yyyymddvv ##### umsCodeGenerator
#####
#####
```

##### 3.1.1.2. #####

##### C ##

```
#####C ##### ISO C 1999 #####
ISO C 1999
```

- #####
- ##### (TBD)

```
gcc -fstd=c99 -Wall ##### int_least16_t # int16_t
#####
```

- [#####1](http://www-6.ibm.com/jp/developerworks/linux/040507/j_1-c99.html) (http://www-6.ibm.com/jp/developerworks/linux/040507/j\_1-c99.html)
- [#####2](http://www.cqpub.co.jp/Interface/column/freesoft/2002/200210/1.htm) (http://www.cqpub.co.jp/Interface/column/freesoft/2002/200210/1.htm)
- [#####3](http://seclan.dll.jp/c99d/c99d00.htm) (http://seclan.dll.jp/c99d/c99d00.htm)

##### Java

- ```
#####
```
- Java2 1.4
 - Java5

[2005051901#Java#C#XSL](#) #####

UMS [ums](#) ##### (####, ####)

#####C#Java #####

#####

3.1.1.3. #####

#####

| #### | ##### | ##### | ## |
|-----------------|--------|-------|------------------------------------|
| .c .h .java .sh | EUC-JP | LF | vi ## emacs
|
| .xml .xsl | UTF-8 | LF | XML#####UTF-8
UTF-16
|

#####

- [###](http://www.muroix.com/charcode/charcode.html) (http://www.muroix.com/charcode/charcode.html)
- [###](http://hokupon.hp.infoseek.co.jp/html/tohoho/wwwkanji.htm) (http://hokupon.hp.infoseek.co.jp/html/tohoho/wwwkanji.htm)
- [###](http://www.ingrid.org/java/i18n/#ji18n) (http://www.ingrid.org/java/i18n/#ji18n)

3.1.1.4. #####

- #####
- #####
- #####

3.1.1.5. #####

\$TABLETOOLS_HOME/changes.xml

- #####(yyyymmddv)
- ###
- #####

#####\$TABLETOOLS_HOME/tool/testAll.sh #####

#####isas/archive ##### umsCodeGenerator-yyyymmddv.tar.gz #####

#####Subversion #####

Warning:
testAll.sh ##### commit #####

umsCodeGenerator

#####

3.1.1.6.

forrest

forrest #####Subversion #####

Warning:

forrest ##### commit #####

3.1.1.7.

[ToDo](#) # [Bugzilla](#)

(http://eda.plain.isas.jaxa.jp/bugzilla/buglist.cgi?product=UMS&bug_status=NEW&bug_status=ASSIGNED)

#####

- ToDo#####
- Bugzilla#####

3.1.1.8.

[Releases](#) #####

- #####
- #####
- ##### bugfix ###(Bugzilla ###Bug ###)#

2006#06#23# ##### [pod](#) (<http://perldoc.jp/docs/perl/5.6.1/perlpod.pod>)

pod #####pod ### yyymmddv_ums_release.pod

pod2html

```
$ pod2html --css=ums.css --title="yyymmddv
umsCodeGenerator release" < yyymmddv_ums_release.pod >
yyymmddv_ums_release.html
```

3.1.2.

3.1.2.1.

umsCodeGenerator #####Java#XSLT #####

| ## | #### | ## |
|-----|------------|---------------------|
| ### | ##### Java | Bourne #####C ##### |

| | | |
|------|---|---|
| | ##### | |
| Java | #####(#####XSLT
####)
XSLT
XSLT ### | |
| XSLT | mapping definition #####
mapping definition ###
mapping definition
| XSLT ##### Xalan-Java
(http://xml.apache.org/xalan-j/)
#####
XSLT ## |

#####

1. #####Java #####
2. #####
3. #####
4. mapping definition #####
5. mapping definition #####
6. UMS ####
7. define/ref ###
8. UMS #####define/ref #####
9. mapping definition ##### XSLT #####
10. mapping definition #####
11. mapping definition # Language syntax ###
12. #####

Note:

#####

#####[build#####](#)

3.1.2.2. 1. #####Java #####

Java

```
[#####]
src/txt2clng.tpl
src/txt2java.tpl
```

3.1.2.3. 2. #####

"txt2lang" #####USAGE #####

#####Jakarta-Commons ##### [CLI](http://jakarta.apache.org/commons/cli/)
(<http://jakarta.apache.org/commons/cli/>) #####

```
[#####]  
src/java/jp/jaxa/isas/ums/codeGenerator/GeneratorMain.java
```

3.1.2.4. 3. #####

#####

"\$TABLETOOLS_TEMP/yyyyMMddHHmmssSSS##(0.0 ## 1.0 ### double #)"

```
[#####]  
src/java/jp/jaxa/isas/ums/codeGenerator/WorkDirectory.java
```

3.1.2.5. 4. mapping definition #####

MSV # Jing #####mapping definition #####
mapping definition
#####

| mapping definition | #### |
|-----------------------|------------------|
| XML syntax(C ##) | dat_xml-clng.rng |
| XML syntax(Java) | dat_xml-java.rng |
| Language syntax(C ##) | dat_lang.rng |
| Language syntax(Java) | dat_lang.rng |

####value ### type ##### ## facet #####
[encodengLibrary](#) #####

```
[#####]  
src/java/jp/jaxa/isas/ums/codeGenerator/XmlValidator.java
```

####[value](#) ##### [facet](#)

3.1.2.6. 5. mapping definition #####

mapping definition ##### mapping definition #####

- #####
- ###
- ###

mapping definition ##### XSLT ##### XSLT

mapping definition

```
[#####]
src/java/jp/jaxa/isas/ums/codeGenerator/XmlEditor.java
```

3.1.2.7. 6. UMS #####

#####UMS #####4.3. datatypeLibrary #####4.4. value ### type ### #####

| ## | ## |
|-------------------------|--|
| 4.3. datatypeLibrary ## | datatypeLibrary ##### data ###value
#####
datatypeLibrary ##### datatypeLibrary
#####
data ###value ##### datatypeLibrary
|
| 4.4. value ### type ## | type ##### value ##### token ### type
#####
datatypeLibrary ##### |

```
[#####]
src/xslt/raw/simplify.xsl
```

3.1.2.8. 7. define/ref ###

umsCodeGenerator #####mapping definition ##### define ### ref
#####

```
[#####]
src/xslt/raw/resolveRef.xsl
```

#####[XML](#) #####

3.1.2.9. 8. UMS #####define / ref #####

UMS #####define /ref ##### # mapping definition
#####

| mapping definition | #### |
|-----------------------|-------------------------|
| XML syntax(C ##) | simple_dat_xml-clng.rng |
| XML syntax(Java) | simple_dat_xml-java.rng |
| Language syntax(C ##) | simple_dat_lang.rng |

| | |
|-----------------------|---------------------|
| Language syntax(Java) | simple_dat_lang.rng |
|-----------------------|---------------------|

```
[#####]  
src/java/jp/jaxa/isas/ums/codeGenerator/XmlValidator.java
```

3.1.2.10. 9. mapping definition ##### XSLT

9.1 ##### XSLT

```
#####*.ums ##### file  
#####
```

```
##### mapping definition #####mapping definition # #####  
XSLT #####Xalan-Java #####
```

```
[#####]  
src/java/jp/jaxa/isas/ums/codeGenerator/FileOutputTableCreator.java
```

#####

9.2 ##### XSLT template

```
#####mapping definition # defineMapping ##### defineMapping  
##### defineMapping #####XSLT #####  
defineMapping #####
```

```
####defineMapping ##### XSLT # template #####
```

```
[#####]  
src/java/jp/jaxa/isas/ums/codeGenerator/VarTypeTableCreatorForCling.java  
src/java/jp/jaxa/isas/ums/codeGenerator/VarTypeTableCreatorForJava.java
```

#####

9.3 XSLT template

```
XSLT #####  
#####XSLT # template #####
```

```
"pre_txt2lang.xsl" ## <xsl:template name="varTypeDefinition"/> ##### 9.2 #####  
template #####
```

```
[#####]  
src/java/jp/jaxa/isas/ums/codeGenerator/Generator.java
```

3.1.2.11. 10. mapping definition #####

[mapping definition](#) ##### UMS #####

- define #####optional #oneOrMore#zeroOrMore #####
- loop ##### 32 #####
- byte ###list ##### bit #####
- bit ### encode ##### "txt" #####
- method #####"2"#"3"#"4" #####
- umsCodeGenerator #####
- ##### UMS_THREAD # "true" #####
- UMS #####7.1.5. start #####

```
[#####]
src/xslt/raw/check_mappingdef.xsl
src/xslt/raw/check.xsl
```

3.1.2.12. 11. mapping definition # Language syntax ###

9.3 ##### XSLT ##### XML syntax # mapping definition ##Language syntax #####
mapping definition # Language syntax #####mapping definition ####

```
[#####]
src/xslt/raw/pre_txt2clng.xsl
src/xslt/raw/pre_txt2java.xsl
```

#####[XML-language Syntax](#) #####

3.1.2.13. 12. #####

#####

1. <post:* /> # file #####
2. <post:* /> #### file #####
3. file #####

#####[XSLT](#) #####pre#####

#####

#####[mapping definition](#) #####

#####[tableTools # umsCodeGenerator](#) ##### [XSLT template](#)

12.1 <post:* /> # file #####

#####

| | |
|-------|----|
| ##### | ## |
|-------|----|

| | |
|--------------|---------------------|
| txt2clng.xml | #####C ##### |
| txt2chdr.xml | #####C #####(#####) |
| txt2java.xml | #####JAVA##### |

"txt2lang.xml" ##### mapping definition ##### XSLT
#####(##### "###.xml")#

XSLT # "txt2clng.xml"#"txt2java" #####
C###JAVA#####
post ##### "txt2lang.xml" ##### XSLT # xsl:include
#####

```
[#####]
src/xslt/raw/txt2clng.xml
src/xslt/raw/txt2chdr.xml
src/xslt/raw/txt2java.xml
```

12.2 <post:* /> #### file #####

C ### Java #####XSLT #####XSLT
#####"post" #####

post #####

- ####
- ###/#####
- ##/#####

#####2#####

| ### | ## |
|--------------|------------------|
| <post:clng/> | ##### C## ##### |
| <post:java/> | ##### JAVA ##### |

```
[#####]
src/xslt/m2/secondTransform_clng.xml
src/xslt/m2/secondTransform_java.xml
src/xslt/m3/secondTransform_java.xml
src/xslt/m4/secondTransform_clng.xml
src/xslt/raw/post_clng.xml
src/xslt/raw/post_java.xml
```

#####[post ##### XSLT ##](#)

12.3 file #####

9.1 ##### XSLT #####

#####

3.1.2.14. [####] #####

umsCodeGenerator #### "-d" #####/#####

- mapping definition #####
- #####
- #####
- #####

#####

3.1.2.15. [####] #####

#####umsCodeGenerator #####
 2006/10/27 ##### 2006/10/27 ##### "false" #####
 #####

[#####]
 src/java/jp/jaxa/isas/ums/codeGenerator/GeneratorMain.java

#####

3.1.3. #### / #####

3.1.3.1. #####

umsCodeGenerator ##2#####
 ##### 2#####

| ##### | ##### | ##### |
|-------|-------|-------|
| ##### | #### | #### |
| ##### | #### | ##### |

Note:
 ##### "####" ##### "####" ##### umsCodeGenerato #####"##" #
 brunch#sleep#awake#sync #####"####" #####

umsCodeGenerator

```
#####  
#####Java##  
#####C####  
#####case #####  
#####
```

3.1.3.2.

umsCodeGenerator #####

| ##### | ## |
|------------|---|
| dat | bit#byte#data#value ##### |
| list | list + dat ### |
| loop | oneOrMore#zeroOrMore#optional + dat ### |
| interleave | interleave + dat ### |

umsCodeGenerator #####dat#list#loop#interleave ## thread #####
#####

```
[#####]  
src/clng/umsDat.h  
src/clng/umsDat.c  
src/clng/umsList.h  
src/clng/umsList.c  
src/clng/umsLoop.h  
src/clng/umsLoop.c  
src/clng/umsInterleave.h  
src/clng/umsInterleave.c  
src/clng/umsThread.h  
src/clng/umsThread.c  
src/java/jp/jaxa/isas/ums/runtime/UMSDat.java  
src/java/jp/jaxa/isas/ums/runtime/UMSList.java  
src/java/jp/jaxa/isas/ums/runtime/UMSLoop.java  
src/java/jp/jaxa/isas/ums/runtime/UMSInterleave.java  
src/java/jp/jaxa/isas/ums/runtime/UMSThread.java
```

```
#####  
#####
```

3.1.3.3.

bit / byte

#####[push, pop # template #####](#)
#####[##### method 2 #### epos #####](#)

data / value

#####[data / value #####](#)
#####[validate -> restrict -> sync -> #####](#)

param

#####[#####](#)

list

#####[list ###](#)
#####[#####\(template # foreach #####\)](#)

choice

#####[choice ###](#)
#####[#####\(template # foreach #####\)](#)

optional

#####[optional ###](#)

oneOrMore

#####[oneOrMore ###](#)

zeroOrMore

#####[zeroOrMore ###](#)

interleave

#####[interleave #####](#)
#####[interleave ###](#)

3.1.3.4. #####

umsCodeGenerator

- mapping function [2005062651](#):mapping function ##### (##)
- [2005062451](#): #####
- [2005062452](#): tableTools # umsCodeGenerator ##
- [2005080301](#): umsCodeGenerator, tableTools ###
- [2005072953](#): ##### (txt:line ###encodingLibrary #####param ## #####)
- [2005070251](#): XSL # XSL # XSL ####
- [2006062201](#): pre ##### XSL ##

3.1.4.

3.1.4.1.

#####2#####

1. mapping definition #####
2. #####/#####

3.1.4.2. 1.

XSLT ##### Java
#####

<xsl:template name="error"> ##### ##
template #####

```
# Error - mapping definition ##### line:### column:###  
element:<###> "#####."
```

XSLT ##### XSLT #####

```
[#####]  
src/raw/common.xsl  
src/raw/check.xsl
```

#####[XSLT](#) #####

3.1.4.3. 2.

umsCodeGeneratotr #####/#####
#####/#####

- #####
- #####
- #####
- ##### mapping definition #####

- ##### mapping definition ###
- ##### mapping definition ###
- ##### mapping definition ###
- #####(bit)
- #####

#####

```
#####
-- input data --
  at line ###, column ###
  at byte ####, bit ####
  Address| 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
  -----+-----
      xxxx | #####
              ~ #####
-- mapping definition --
  mapping definition #####
  at line:###, column:###, element:<###>, "#####"
  ###
  at line:###, column:###, element:<###>, "#####"
  ###
  :
```

Note:
#####(#####)## #####

2.1 #####

###/#####C ### Java ##### Java #####C
##Java

[#####](#)

2.1.1 C #####

C##### "ums__exception_t" #####

```
[#####]
src/clng/umsException.h
```

Java #####

"FUNCTION" ##

| ### | ## |
|------------------------|-------------------------------------|
| throw | ##### void ##### |
| throwWithValue | #####
void##### return ### |
| catchAndThrow | ##### void ##### |
| catchAndThrowWithValue | ##### void
return ### |
| catch | ##### |

```
[#####]
src/clng/umsException.h
```

2.1.2 Java #####

Java ###Exception ##### UMSEException #####

```
[#####]
src/java/jp/jaxa/isas/ums/runtime/UMSEException.java
```

Java #####

#####[JAVA](#) ##### clone() #####

2.2 #####

#####10#6#####

```
WWXYZZ
WW: ####
X ####
Y #####
ZZ###
```

#####umsCodeGenerator #####

| # | ##### | ## |
|----|-------|------------------|
| WW | 10 | ##### |
| | 20 | ##### |
| | 30 | ##### |
| X | 1 | umsCodeGenerator |

| | | |
|---|---|-------------------|
| Y | 1 | datatypeLibrary## |
| | 2 | decode#encode## |
| | 3 | ##### |
| | 4 | ##### |

umsCodeGenerator#####

| ### | ##### | ##### | ##### |
|-------|-------------------|--------|-------------|
| ##### | datatypeLibrary## | 101101 | ###bit##### |
| ##### | datatypeLibrary## | 101103 | ###bit##### |
| ##### | decode#encode## | 101209 | ##### |
| ##### | decode#encode## | 101211 | ##### |
| ##### | datatypeLibrary## | 201104 | ##### |
| ##### | decode#encode## | 201204 | ##### |
| ##### | decode#encode## | 201205 | ##### |
| ##### | decode#encode## | 201206 | ##### |
| ##### | decode#encode## | 201900 | ##### |
| ##### | datatypeLibrary## | 301100 | ##### |
| ##### | datatypeLibrary## | 301102 | ##### |
| ##### | datatypeLibrary## | 301105 | ##### |
| ##### | datatypeLibrary## | 301172 | ##### |
| ##### | decode#encode## | 301200 | ##### |
| ##### | ##### | 301300 | bit##### |

2.3 #####

#####

#####=#####

#####

| | | |
|----|-------|-------|
| ## | ##### | ##### |
|----|-------|-------|

| | | |
|-----|----------------------------|--|
| ### | UMSException_jp.properties | 101101=#####length#####(bitlen)#encode#####(typelen)#####
#####length#####
101103=#####length#####(bitlen)#####
#####(maxlen)#####
#####length#####
|
| ## | UMSException_en.properties | 101101=It is not equal to fixed bit length (typelen) of the data type from which bit length (bitlen) specified for length of the container is specified for encode.
Please correct the value specified for length of the container.
101103=Bit length (bitlen) specified for length of the container exceeds maximum bit length (maxlen)of the data type. Please correct the value specified for length of the container or change the data type.
|

```
[#####]
lang/UMSException_en_US.properties
lang/UMSException_ja_JP.properties
```

Note:

```
##### LANG #####
```

3.1.5. Type Conversion Library

3.1.5.1. 1. Type Conversion Library (####)

```
UMS ###RELAX NG #####
```

- encode##datatype
- datatype#####

```
##### ### Type Conversion Library ##### umsCodeGenerator ##### XSLT #####
```

```
##### umsCodeGenerator ### XSLT ## ##### Type Conversion Library #####
umsCodeGenerator ### XSL ##datatypeLibrary ##### 'library'
#####template ##### template #'library' ##### # Type Conversion Library
## XSLT ##### Type Conversion Library #####
### Type Conversion Library ##### XSLT #####
##### (#####)
```

1.1 typeConversion_lang.xml

```
<stylesheet>
  <include href="file://localhost/W3C_typeConversion_lang.xml" />

  <template name="initLibrary" />

  <template name="encode2valuetypeConversion"
    param="library, datatype, encode, cont10, cont20,
constant"/>
  <template name="encode2datatypeConversion"
    param="library, datatype, encode, cont10, cont20"/>
  <template name="datatype2langtypeConversion"
    param="library, datatype, langtype, varname"/>
  <template name="langtype2datatypeConversion"
    param="library", datatype, langtype, varname"/>
  <template name="datatype2encodeConversion"
    param="library, datatype, encode, cont10, cont20"/>
  <template name="valuetype2encodeConversion"
    param="library, datatype, encode, cont10, cont20,
constant"/>
</stylesheet>
```

1.2 typeConversion_chdr.xml

```
<stylesheet>
  <include href="file://localhost/W3C_typeConversion_chdr.xml" />

  <template name="useLibrary" />
</stylesheet>
```

3.1.5.2. 2. Type Conversion Library (#####)

```
##### Type Conversion Library ##### umsCodeGenerator
##### W3C #####

#### template ##### datatype ##### data #### value ### type
##### umsCodeGenerator ### XSL ##### # type conversion library
```

```
##### #### datatype #####  
## XSL #####  
<value /> #####  
##### type ##### XSL ##### JAVA #####
```

2.1 W3C_typeConversion_lang.xml

```
<stylesheet>  
  <template name="W3C_initLibrary"/>  
  <template name="W3C_dataTypeDefinition"  
    param="datatype"/>  
  <template name="W3C_encode2valuetypeConversion"  
    param="datatype, encode, cont10, cont20, constant"/>  
  <template name="W3C_encode2datatypeConversion">  
    param="datatype, encode, cont10, cont20"/>  
  <template name="W3C_datatype2langtypeConversion">  
    param="datatype, langtype, varname"/>  
  <template name="W3C_langtype2datatypeConversion">  
    param="datatype, langtype, varname"/>  
  <template name="W3C_datatype2encodeConversion">  
    param="datatype, encode, cont10, cont20"/>  
  <template name="W3C_valuetype2encodeConversion">  
    param="datatype, encode, cont10, cont20, constant"/>  
</stylesheet>
```

2.2 W3C_typeConversion_chdr.xml

```
<stylesheet>  
  <template name="W3C_useLibrary"/>  
</stylesheet>
```

3.1.5.3. 3. Type Conversion Library (Library Function)

```
#####  
#####  
#####umsCodeGenerator#####  
##### typeConversionLibrary # #####
```

3.1 W3C_typeConversionLibrary.{h,c}

```
void ums__W3C_typeConversionLibrary_init(void);
```

```

void ums__W3C_typeConversionLibrary_end(void);

char *ums__W3C_txtEncode_stringType
    (ums__dat_t *dat, int epos, ums__exception_t *ex);
char *ums__W3C_txtEncode_tokenType
    (ums__dat_t *dat, int epos, ums__exception_t *ex);
int32_t ums__W3C_txtEncode_intType
    (ums__dat_t *dat, int epos, ums__exception_t *ex);
uint32_t ums__W3C_txtEncode_unsignedIntType
    (ums__dat_t *dat, int epos, ums__exception_t *ex);
double ums__W3C_txtEncode_doubleType
    (ums__dat_t *dat, int epos, ums__exception_t *ex);
int32_t ums__W3C_signedEncode_intType
    (ums__dat_t *dat, int epos, ums__exception_t *ex);
uint32_t ums__W3C_unsignedEncode_unsignedIntType
    (ums__dat_t *dat, int epos, ums__exception_t *ex);
double ums__W3C_ieee754doubleEncode_doubleType
    (ums__dat_t *dat, int epos, ums__exception_t *ex);

void ums__W3C_stringType_txtEncode
    (char * str, ums__dat_t *dat, int epos, ums__exception_t *ex);
void ums__W3C_tokenType_txtEncode
    (char * str, ums__dat_t *dat, int epos, ums__exception_t *ex);
void ums__W3C_intType_txtEncode
    (int32_t num, ums__dat_t *dat, int epos, ums__exception_t *ex);
void ums__W3C_unsignedIntType_txtEncode
    (uint32_t num, ums__dat_t *dat, int epos, ums__exception_t *ex);
void ums__W3C_doubleType_txtEncode
    (double num, ums__dat_t *dat, int epos, ums__exception_t *ex);
void ums__W3C_intType_signedEncode
    (int32_t num, ums__dat_t *dat, int epos, ums__exception_t *ex);
void ums__W3C_unsignedIntType_unsignedEncode
    (uint32_t num, ums__dat_t *dat, int epos, ums__exception_t *ex);
void ums__W3C_doubleType_ieee754doubleEncode
    (double num, ums__dat_t *dat, int epos, ums__exception_t *ex);

void ums__W3C_txtEncode_stringValue
    (ums__dat_t *dat, int epos, char *literal, ums__exception_t *ex);
void ums__W3C_txtEncode_tokenValue
    (ums__dat_t *dat, int epos, char *literal, ums__exception_t *ex);
void ums__W3C_txtEncode_intValue
    (ums__dat_t *dat, int epos, int32_t literal, ums__exception_t *ex);
void ums__W3C_txtEncode_unsignedIntValue
    (ums__dat_t *dat, int epos, uint32_t literal, ums__exception_t *ex);
void ums__W3C_txtEncode_doubleValue
    (ums__dat_t *dat, int epos, double literal, ums__exception_t *ex);
void ums__W3C_signedEncode_intValue
    (ums__dat_t *dat, int epos, int32_t literal, ums__exception_t *ex);
void ums__W3C_unsignedEncode_unsignedIntValue
    (ums__dat_t *dat, int epos, uint32_t literal, ums__exception_t *ex);
void ums__W3C_ieee754doubleEncode_doubleValue
    (ums__dat_t *dat, int epos, double literal, ums__exception_t *ex);

```

3.2 Type Conversion Library (Library Function)

```
public class W3C_TypeConversionLibrary {  
    public static String txtRead(UMSDat dat, int epos)  
        throws UMSEException;  
    public static String txtEncode_stringType(UMSDat dat, int epos)  
        throws UMSEException;  
    public static String txtEncode_tokenType(UMSDat umsDat, int epos)  
        throws UMSEException;  
    public static int txtEncode_intType(UMSDat umsDat, int epos)  
        throws UMSEException;  
    public static double txtEncode_doubleType(UMSDat umsDat, int epos)  
        throws UMSEException;  
    public static int signedEncode_intType(UMSDat umsDat, int epos)  
        throws UMSEException;  
    public static double ieee754doubleEncode_doubleType(UMSDat umsDat, int  
epos)  
        throws UMSEException;  
    public static void bitWrite(long num, UMSDat umsDat, int epos)  
        throws UMSEException;  
    public static void stringType_txtEncode(String str, UMSDat umsDat, int  
epos)  
        throws UMSEException;  
    public static void tokenType_txtEncode(String str, UMSDat umsDat, int  
epos)  
        throws UMSEException;  
    public static void intType_txtEncode(int num, UMSDat umsDat, int epos)  
        throws UMSEException;  
    public static void doubleType_txtEncode(double num, UMSDat umsDat, int  
epos)  
        throws UMSEException;  
    public static void intType_signedEncode(int num, UMSDat umsDat, int epos)  
        throws UMSEException;  
    public static void doubleType_ieee754doubleEncode(double num, UMSDat  
umsDat, int epos)  
        throws UMSEException;  
    public static void txtEncode_stringValue(UMSDat dat, int epos, String  
literal)  
        throws UMSEException;  
    public static void txtEncode_intValue(UMSDat dat, int epos, int literal)  
        throws UMSEException;  
    public static void txtEncode_doubleValue(UMSDat dat, int epos, double  
literal)  
        throws UMSEException;  
    public static void signedEncode_intValue(UMSDat dat, int epos, int  
literal)  
        throws UMSEException;  
    public static void ieee754doubleEncode_doubleValue(UMSDat dat, int epos,  
double literal)  
        throws UMSEException;  
}
```

3.1.6. #####

3.1.6.1. #####

(-- ##### --)

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

1. #####
2. mapping definition ###
3. #####
4. #####
5. #####

```
##### mapping definition #####UMS#####
##### mapping
definition #####
```

3.1.6.2. 1. #####

```
#####mapping definition #####
#####UMS#####
```

3.1.6.3. 2. mapping definition ###

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

```
<grammar xmlns="http://ums.isas.jaxa.jp/0.4">
  <start>
    <java:class scope="abstract" name="XmlSample"
xmlns:java="http://ums.isas.jaxa.jp/0.4/java">
      <defineVariables>
        <java:var name="rec" class="Record"/>
      </defineVariables>
    </java:class>
  </start>
</grammar>
```

Note:
 #####trang #####schema/*.*.rng #compact syntax ##### Emacs/nxml-mode ##### mapping definition #
 validate #####

3.1.6.4. 3. #####

```
##### decode#encode ##### mapping definition
##### mapping definition ##### XSLT
##### datatypeLibrary#####
##### main #####
#####
#####
```

3.1.6.5. 4. #####

```
##### mapping definition ##### mapping definition
##### XSLT ##### mapping
definition # XSL #####
##### RELAX NG ##### XSLT #####
```

```
<element name="a" />
# <xsl:foreach select="a" />
#####a#####XSLT #a#####
###varidation#####
```

```
<zeroOrMore>
  <element name="a" />
</zeroOrMore>
# <xsl:foreach select="a" />
```

```
<optional>
  <attribute name="@a" />
</optional>
# <xsl:if test="@a">####</xsl:if>
```

```
<choice />
# <xsl:choose />
```

```
<define name="xxx" /> # <ref name="xxx" /> ##
# <template name="xxx" /> # <call-template name="xxx" /> ##
```

XSLT

```
<element name="a">
  <element name="b" />
  <element name="c" />
</element>
# <xsl:for-each select="a"><xsl:for-each
```

```

select="b" /></xsl:for-each>
# <xsl:for-each select="a"><xsl:for-each
select="c" /></xsl:for-each>

```

XSLT

```

<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:fo="http://www.w3.org/1999/XSL/Format"
xmlns:ums="http://ums.isas.jaxa.jp/0.4"
xmlns:java="http://ums.isas.jaxa.jp/0.4/java">
  <xsl:output method="text"/>
  <xsl:template match="/">
    <xsl:for-each select="ums:grammar">
      <xsl:for-each select="ums:start">
        <xsl:for-each select="java:class"><xsl:text></xsl:text>
          <xsl:if test="@scope">
            <xsl:value-of select="@scope"/>
            <xsl:text> </xsl:text>
          </xsl:if>class <xsl:value-of select="@name"/> {
          <xsl:for-each select="ums:defineVariables">
            <xsl:for-each select="java:var">
              <xsl:choose>
                <xsl:when test="@type">
                  <xsl:value-of select="@type"/>
                </xsl:when>
                <xsl:when test="@class">
                  <xsl:value-of select="@class"/>
                </xsl:when>
              </xsl:choose>
              <xsl:text> </xsl:text>
              <xsl:value-of select="@name"/>;
            </xsl:for-each>
          </xsl:for-each>
        }
      </xsl:for-each>
    </xsl:for-each>
  </xsl:template>
</xsl:stylesheet>

```

3.1.6.6. 5.

```

##### XSLT ##### XSLT #####mapping definition #####
#####

```

3.2. Todo List

3.2.1. To do

See <http://eda.plain.isas.jaxa.jp/bugzilla> for bug tracking.

Following tasks are listed in lower priority.

3.2.1.1. medium

- [code] #####
- [code] List #####(epos # Thread #####pushContainer #####)
- [code] Interleave #####

3.2.1.2. low

- [code] XML version
- [code] Support C++
- [code] Support Perl
- [code] Recursive version
- [docs] Support of restriction of output only element
- [docs] howto #####
- [code] #####
- [code] #####
- [code] #####
- [code] Relax NG#####(#####)

3.2.1.3.

- ##### ums__ #### UMS__ ##### ##umsCodeGenerator, tableTools #### tableTools_init() #####
- #####typeConversionLibrary#####
- #####
- encodingLibrary # XSL #####template ## param ##### #####
- ant ## ant #####(cygwin)#####ant ## ant #####

3.3. Related Information for usage and development of mappingSchema

3.3.1. SSH

- SSH ##### RSA ##

3.3.2. Subversion

- Subversion #####(Japanese)
- Subversion #####(Japanese)

3.3.3. XML Editor

nxml mode in Emacs

- Emacs/nXML(Japanese)
- nxml-mode(Japanese)

psgml mode in Emacs

- Usage of PSGML mode(Japanese)
- psgml-mode(Japanese)

XMLspy

- Altova
- Toshiba Solutions(Japanese)

3.4. Others

3.4.1. Sibling of mappingSchema under discussion

3.4.1.1. Timing

[Requirement](#) of a data processor

[Design Results](#) of this data processor

Example of [Parallel Processing](#) (Variable period processing + constant period processing)

3.4.1.2. Mail Archive

[20040125](#)

3.4.2. xml2fits

3.4.2.1. Page for XML2FITS Developers and Users

What is XML2FITS

XML2FITS is a configurable format conversion tool from XML format to FITS format with binary extensions (Figure 1). Structure of input data of this tool is specified by an XML schema language -- RELAX NG --and that of output is also specified by a schema described in XML. This tool is configured by a file named Mapping Schema created from the input schema and the output schema.

Figure 1

ADASS2004

- Poster
- Proceedings

3.4.2.2. Installation of XML2FITS

Installation

.

Required Softwares

Following software is required for usage of the program.

- [J2SE 1.4.2 \(J2SE\)](http://java.sun.com/) (http://java.sun.com/) (Environment Variable: JAVA_HOME; [installation memo](#))
- [Apache Ant 1.6](http://ant.apache.org/) (http://ant.apache.org/) (Environment Variable: ANT_HOME; [installation memo](#))
- [CFITSIO 2.5](http://heasarc.gsfc.nasa.gov/docs/software/fitsio/fitsio.html) (http://heasarc.gsfc.nasa.gov/docs/software/fitsio/fitsio.html) ([installation memo](#))
- [JUnit 3.8](http://www.junit.org/index.htm) (http://www.junit.org/index.htm) ([installation memo](#))
- [Jakarta log4j 1.2](http://logging.apache.org/) (http://logging.apache.org/) (included in the distribution, no individual install need)
- [Jakarta Commons Logging 1.0](http://jakarta.apache.org/commons/logging/) (http://jakarta.apache.org/commons/logging/) (included in the distribution, no individual install need)
- [msv](http://www.sun.com/software/xml/developers/multischema/) (http://www.sun.com/software/xml/developers/multischema/)
- [jing](http://www.thaiopensource.com/relaxng/jing.html) (http://www.thaiopensource.com/relaxng/jing.html)

All of the executable of the software above should be included into the command path (eg. Environment Variable: PATH of shell).

3.4.2.3. Sample usage of umsCodeGenerator

Sample Files

Schema of Input mapping definition	xmlfits-input-20040825.r	Input Data	xmlfits-input-20040825.xml
Schema of Output	xmlfits-input-20040825-f	Output Data	xmlfits-input-20040825.head (only FITS header)

A **mapping definition file** can be created from **schema of input** and **schema of output**. On the other hand, **mapping definition file** can be split into **schema of input (description of external data)** and **schema of output (description of interface with program language)**.

3.4.2.4. Reference

FITS

- Tutrial in Japanese <http://www.fukuoka-edu.ac.jp/~kanamitu/fits/tebiki2/1-gaiyou.html>

4. All