

M2TK解析模板定义说明

目录

- 一、介绍
 - 1.1 文档编写目的
 - 1.2 文档面向人群
- 二、模板定义
 - 2.1 XML声明
 - 2.2 <M2TKTemplate> 标签
 - 2.3 <TableTemplate> 标签
 - 2.3.1 <TableId> 标签
 - 2.3.2 <DisplayName> 标签
 - 2.3.3 <TableBody> 标签
 - 2.3.4 <UniqueKey> 标签
 - 2.4 <DescriptorTemplate> 标签
 - 2.4.1 <DisplayName> 标签
 - 2.4.2 <MayOccurIn> 标签
 - 2.4.3 <DescriptorBody> 标签
 - 2.5 <Syntax> 标签
 - 2.5.1 <Field> 标签
 - 2.5.1.1 <FieldPresentation> 标签
 - 2.5.1.2 <Label> 标签
 - 2.5.1.3 <ValueMapping> 标签
 - 2.5.2 <If> 标签
 - 2.5.2.1 单值比较标签 (<CompareWithConst>)
 - 2.5.2.2 多值比较标签 (<CompareWithConstMulti>)
 - 2.5.3 <Loop> 标签
 - 2.5.3.1 <LoopPresentation> 标签
 - 2.5.3.2 <Body> 标签
 - 2.5.4 <Descriptor> 标签

一、介绍

1.1 文档编写目的

本文档将详细介绍构造M2TK解析模板的步骤，以及M2TK解析模板的各元素含义和使用约束，帮助读者写出正确的数据解析模板。

1.2 文档面向人群

本文档主要面向 **M2TK码流分析助手** 的高级用户，即需要编写自定义数据解析器的人员。因此要求读者必须了解 **MPEG-2数据段 (private_section)** 和 **描述符 (descriptor)** 的定义方式，具备基本的XML语法知识（至少知道怎么组织XML结构），以及基础的Java（或其他类似的编程语言）语法知识，知道或熟悉打印函数（printf）的格式化模板用法，例如 **%d**、**%02x** 等。

二、模板定义

2.1 XML声明

```
<?xml version = "1.0" encoding = "UTF-8"?>
<M2TKTemplate>
    ...
</M2TKTemplate>
```

2.2 <M2TKTemplate> 标签

```
<M2TKTemplate xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               xsi:noNamespaceSchemaLocation="/schema/M2TKTemplate.xsd">
    <!-- body -->
</M2TKTemplate>
```

<M2TKTemplate>标签是模板XML的根节点，其包含一个或多个下列标签：

- <TableTemplate>
- <DescriptorTemplate>

标签可以以任意顺序排列，但不允许嵌套。

【说明】 **M2TKTemplate.xsd** 文件定义了模板XML的格式规范。

2.3 <TableTemplate> 标签

```
<!-- TableTemplate示例 -->
<TableTemplate name="program_association_section" group="PSI/PAT">
    <TableId id="0">
        <DisplayName str="PAT"/>
    </TableId>
    <TableBody>
        <Field name="table_id" length="8" encoding="uimsbf"/>
        <Field name="section_syntax_indicator" length="1" encoding="bslbf"/>
        <Field name="0" length="1" encoding="bslbf"/>
        <Field name="reserved" length="2" encoding="bslbf"/>
        <Field name="section_length" length="12" encoding="uimsbf"/>
        <Field name="transport_stream_id" length="16" encoding="uimsbf"/>
        <Field name="reserved" length="2" encoding="bslbf"/>
        <Field name="version_number" length="5" encoding="bslbf"/>
        <Field name="current_next_indicator" length="1" encoding="bslbf"/>
        <Field name="section_number" length="8" encoding="uimsbf"/>
        <Field name="last_section_number" length="8" encoding="uimsbf"/>
        <Loop name="program_association_loop" length_field="implicit"
length_correction="-4">
            <LoopPresentation>
                <NoLoopHeader/>
```

```

        <LoopEmpty str="无节目描述"/>
        <LoopEntry>
            <Prefix str="节目映射"/>
        </LoopEntry>
    </LoopPresentation>
    <Body>
        <Field name="program_number" length="16" encoding="uimsbf">
            <FieldPresentation>
                <Prefix str="节目号"/>
            </FieldPresentation>
        </Field>
        <Field name="reserved" length="3" encoding="bslbf"/>
        <If>
            <Condition>
                <CompareWithConst field="program_number" comp_op="equals"
const="0"/>
            </Condition>
            <Then>
                <Field name="network_pid" length="13" encoding="uimsbf">
                    <FieldPresentation>
                        <Prefix str="NIT PID"/>
                        <Format str="0x%04X"/>
                    </FieldPresentation>
                </Field>
            </Then>
            <Else>
                <Field name="program_map_pid" length="13"
encoding="uimsbf">
                    <FieldPresentation>
                        <Prefix str="PMT PID"/>
                        <Format str="0x%04X"/>
                    </FieldPresentation>
                </Field>
            </Else>
        </If>
    </Body>
</Loop>
    <Field name="CRC_32" length="32" encoding="uimsbf"/>
</TableBody>
<UniqueKey>
    <FieldRef field="table_id"/>
    <FieldRef field="transport_stream_id"/>
    <FieldRef field="version_number"/>
    <FieldRef field="section_number"/>
</UniqueKey>
</TableTemplate>

```

<TableTemplate>（段模板）标签用于描述私有段特征、语法结构和显示要求。

<TableTemplate>标签要求按照顺序依次描述：

- <TableId>标签
- <TableBody>标签

- `<UniqueKey>` 标签

其中，允许出现多个`<TableId>`标签（用于描述具有不同`TableId`的分段，例如EIT），可以省略`<UniqueKey>`标签（每个解析到的分段数据都独立保存）。

`<TableTemplate>` 标签包含下列属性：

- `name`：当前段结构的索引名称，可用于描述符定位（详见描述符模板中的`MayOccurIn`标签）。
- `group`：展示分段结构时，节点所在的结构树的分支名称。

【要求】`name`是必要属性，其他是可选属性。

2.3.1 `<TableId>` 标签

```
<TableId id="0">
  <DisplayName str="PAT"/>
</TableId>
```

每个`<TableId>`标签描述一个特定的`table_id`值。它可以携带一个`<DisplayName>`标签，用来定义对应表的表头名字。

`<TableId>` 标签包含唯一必要属性：

- `id`：`table_id`值，可以填写十进制数（非负整数）或以`'0x'`开头的十六进制数。

2.3.2 `<DisplayName>` 标签

```
<DisplayName str="PAT"/>
```

`<DisplayName>` 标签描述表数据的显示名称，结构与`<Label>` 标签相同，解析器以字面量方式展示文字内容，不进行格式化。

2.3.3 `<TableBody>` 标签

```
<TableBody>
  ...
</TableBody>
```

`<TableBody>` 标签的结构类型详见`Syntax` 标签。

2.3.4 `<UniqueKey>` 标签

```
<UniqueKey>
  <FieldRef field="table_id"/>
  <FieldRef field="transport_stream_id"/>
```

```

    <FieldRef field="version_number"/>
    <FieldRef field="section_number"/>
</UniqueKey>

```

<UniqueKey>标签可以包含一个或以上<FieldRef>标签。<FieldRef>标签只有一个必要属性：field，表示被引用字段的索引名（详见<Field>标签描述）。

<UniqueKey>标签是可选标签。

2.4 <DescriptorTemplate> 标签

```

<!-- DescriptorTemplate示例 -->
<DescriptorTemplate tag="9" name="ca_descriptor">
  <DisplayName str="CA Descriptor"/>
  <MayOccurIn table="conditional_access_section"/>
  <MayOccurIn table="program_map_section"/>
  <DescriptorBody>
    <Field length="8" name="descriptor_tag" encoding="uimsbf">
      <FieldPresentation>
        <Prefix str="Tag"/>
        <Format str="0x%02X"/>
      </FieldPresentation>
    </Field>
    <Field length="8" name="descriptor_length" encoding="uimsbf"/>
    <Field length="16" name="CAS_ID" encoding="uimsbf">
      <FieldPresentation>
        <Format str="%04x (%s)"/>
      </FieldPresentation>
    </Field>
    <Field length="3" name="reserved" encoding="bslbf"/>
    <Field length="13" name="CA_PID" encoding="uimsbf">
      <FieldPresentation>
        <Prefix str="ECM PID"/>
        <Format str="0x%04X"/>
      </FieldPresentation>
    </Field>
    <Field length="n/a" name="private_data" encoding="octets"
length_field="implicit"/>
  </DescriptorBody>
</DescriptorTemplate>

```

<DescriptorTemplate>（描述符模板）标签用于描述描述符特征、语法结构和显示要求。

<DescriptorTemplate>标签要求按照顺序依次描述<DisplayName>、<MayOccurIn>和<DescriptorBody>标签。其中，允许出现任意数量的<MayOccurIn>标签（标记允许携带该描述符的表）。

<DescriptorTemplate>标签包含下列属性：

- tag: descriptor_tag，其中保留值'0'用于表示未知格式的任何描述符解析。可以填写十进制数（非负整数）或以'0x'开头的十六进制数。

- tag_ext：扩展descriptor_tag。可以填写十进制数（非负整数）或以'0x'开头的十六进制数。
- name：描述符索引名称。

【要求】tag、name是必要属性，其他是可选属性。

2.4.1 <DisplayName> 标签

```
<DisplayName str="CA Descriptor"/>
```

<DisplayName>标签描述描述符的显示名称，结构与<Label>标签相同，解析器以字面量方式展示文字内容，不进行格式化。

2.4.2 <MayOccurIn> 标签

```
<MayOccurIn table="program_map_section"/>
```

<MayOccurIn>标签只有一个必要属性：table，表示可能携带该描述符的表的索引名称。

<MayOccurIn>标签是可选标签。当未定义MayOccurIn时，描述符可以出现在任意表中。

2.4.3 <DescriptorBody> 标签

```
<DescriptorBody>
...
</DescriptorBody>
```

<DescriptorBody>标签的结构类型详见Syntax标签。

2.5 <Syntax> 标签

Syntax标签表示由下列标签任意组合而成的标签集合，包括：

- Field标签：描述具体的单项数据字段。
- If标签：描述在某些情况下存在的数据字段（或字段组合）。
- Loop标签：描述循环出现的数据字段（或字段组合）。
- Descriptor标签：描述符占位符。

解码器按照Syntax标签里元素定义的顺序进行解码，对于字段引用，解码器在当前字段的前序字段（即已出现的字段）中查找引用本体。对于循环中出现的引用，则仅在【当前循环内】和【整个循环体】的前序字段中寻找引用本体。

2.5.1 <Field> 标签

```
<!-- Field示例 -->
<Field name="descriptor_tag" encoding="uimsbf" length="8">
  <FieldPresentation>
    <Prefix str="Tag"/>
    <Format str="0x%02X"/>
  </FieldPresentation>
</Field>
```

<Field>标签描述具有完整含义的简单数据，例如数值、字符串或字节数组，其在含义上完整不可分割（不允许进行拆分解释）。<Field>标签包含以下属性：

- name：字段名称
- encoding：编码方式，包括：bslbf（比特流，最高位在前），uimsbf（无符号整数，最高位在前），checksum（校验码，高位在前），nibbles（半字节流），octets（字节流），text（文本）。
- string_type：文本类型，包括：dvb_text（DVB方式编码的字符串），utf16/utf8，ascii，gb2312/gbk/gb18030（中文编码）。
- length：字段长度，表示字段的编码长度（单位：比特）。通常用于描述数值，或较短的字节流、半字节流。有效范围：[1, 64]。
- length_field：长度引用字段，表示当前字段的编码长度需要用被引用的字段的值间接表示（单位：字节）。当字段长度不固定，但会一直延续到编码末尾时，可以用length_field="implicit"表示。
- length_correction：长度修正值（单位：字节），表示当前字段的实际长度需要进行额外的修正，仅在字段长度为【implicit】时有意义。修正值通常是负数，表示需要减去的字节数。

【例子】PAT关联描述循环是不定长的（一直延续到分段末尾的CRC32字段之前），并且没有明确的长度字段，所以循环的间接长度（分段总长减去起始位置）需要额外减去CRC32字段长度（length_correction=-4）。

【要求】name、encoding是必要属性，其他是可选属性。当字段类型为text时，必须提供合适的string_type描述。

bslbf、**uimsbf**、**checksum** 都是数值类型，可以表示64位以内的任意非负整数；**nibbles** 通常表示BCD编码；**octets** 表示任意长度的字节数组；**text** 表示字符串编码。字符串编码中是否包含结尾字符'\0'，由业务层规定，模板解析时并不作说明。

uimsbf 类型要求数值向右对齐，即数值的最低位在字节的末尾。

checksum 类型在 **uimsbf** 类型要求的基础上，增加了对长度的限制，即数值长度仅允许为8、16、32、64位（符合常见CRC校验算法长度）。

<Field>标签可以添加显示说明标签（<FieldPresentation>），专门描述字段的展示形式和特殊要求。

2.5.1.1 <FieldPresentation> 标签

<FieldPresentation> 标签包含两个可选的 **Label** 标签和一个可选的 **ValueMapping** 标签：

- <Prefix> 标签：展示字段时，需要添加在内容前面的前缀部分。默认以当前字段的名称加冒号作为前缀显示。
- <Format> 标签：展示字段时，字段值的显示样式。
- <Mapping> 标签：值映射。

2.5.1.2 <Label> 标签

Label 标签包含以下属性：

- str：表示要显示的文字或格式化模板（仅对Format标签适用）。

<Format> 标签里的格式化模板，需要遵循：

- 当应用于数值时，应符合Java语言里的字符串格式化模板，模板可以接收多个输入，但所有输入值都是字段的值。例如：str="0x%02X (%d)"，节点值将传两遍给格式化模板。
- 当应用于nibbles流时，可以添加格式化说明：compact、lowercase（顺序任意，逗号分隔）或包含“#”符号的BCD格式化模板。compact表示用字符串形式展示（默认是数组形式），lowercase表示字符以小写形式展示（默认大写）。
- 当应用与octets流时，可以添加格式化说明：compact、lowercase（顺序任意，逗号分隔）。compact表示以连续十六进制数字字符串表示，且无前后括号包围（默认以数组形式展示），lowercase表示字符以小写形式展示（默认大写）。

【要求1】 格式化标签不适用于文本内容。

【要求2】 BCD模板只能单独使用，且“#”符号的数量必须与半字节数量一致，否则会导致渲染失败。模板示例如下：

```
<!-- BCD格式模板的示例 -->
<Field name="symbol_rate" length="7" encoding="nibbles">
  <FieldPresentation>
    <Prefix str="符号率"/>
    <Format str="###.#### Msymbol/s"/>
  </FieldPresentation>
</Field>
```

2.5.1.3 <ValueMapping> 标签

```
<!-- ValueMapping示例 -->
<Mapping>
  <Value value="123">
    <ValString str="123"/>
  </Value>
  <ValueRange min="0" max="10">
    <ValString str="123"/>
  </ValueRange>
</Mapping>
```

<ValueMapping> 标签包含下列可选的子标签：

- <Value> 标签：按照单值转义，包含一个属性（value），一个子标签（ValString，Label类型）。
- <ValueRange> 标签：按照区间值（[min,max]）转义，包含两个属性（min：区间下限，包含；max：区间上限，包含），一个子标签（ValString，Label类型）。

- `<DVBTTime>` 标签：按照MJD-UTC格式的转义时间，无属性，无子标签。
- `<Duration>` 标签：按照hh:mm:ss格式转义事件持续时长，无属性，无子标签。
- `<ThreeLetterCode>` 标签：按照3位ASCII编码格式转义语言代码或国家代码，无属性，无子标签。

【注意】 **DVBTTime**、**Duration**、**ThreeLetterCode** 有专门用途，不要与其他标签混用。

Value 标签里的value属性以及 **ValueRange** 标签里的min/max属性，可以填写十进制数（非负整数）或以'0x'开头的十六进制数。

2.5.2 `<If>` 标签

```
<!-- If示例 -->
<If>
  <Condition>
    <CompareWithConst field="program_number" comp_op="equals" const="0"/>
  </Condition>
  <Then>
    <Field name="network_pid" length="13" encoding="uimsbf">
      <FieldPresentation>
        <Prefix str="NIT PID"/>
        <Format str="0x%04X (%d) "/>
      </FieldPresentation>
    </Field>
  </Then>
  <Else>
    <Field name="program_map_pid" length="13" encoding="uimsbf">
      <FieldPresentation>
        <Prefix str="PMT PID"/>
        <Format str="0x%04X (%d) "/>
      </FieldPresentation>
    </Field>
  </Else>
</If>
```

`<If>` 标签表示根据某个条件而动态存在的可选结构定义。当比较结果为真（true）时，`<Then>` 标签中的字段生效；当比较结果为假（false）时，`<Else>` 标签中的字段生效。

`<Then>` 标签和 `<Else>` 标签里可以携带一个或以上的Syntax标签。`<Then>` 标签为必要标签，`<Else>` 标签为可选标签。

【注意】待比较对象（字段）应是数值类型，否则会造成解码错误。

目前条件判断为对某个字段的值进行比较操作，比较的方法可以是：

- 【单值比较】等于（equals）
- 【单值比较】不等于（not_equals）
- 【单值比较】大于（larger_then）
- 【单值比较】小于（smaller_then）
- 【多值比较】等于其中某值（equals_any）
- 【多值比较】不等于指定值（not_equals_all）

const属性可以填写十进制数（非负整数）或以'0x'开头的十六进制数。

2.5.2.1 单值比较标签 (<CompareWithConst>)

```
<CompareWithConst field="program_number" comp_op="equals" const="0"/>
```

包括以下属性：

- field：待比较的字段名
- comp_op：比较方法
- const：比较值

2.5.2.2 多值比较标签 (<CompareWithConstMulti>)

```
<CompareWithConstMulti field="program_number" comp_op="equals_any">
  <ConstValue const="0"/>
  <ConstValue const="1"/>
  ...
</CompareWithConstMulti>
```

包括以下属性：

- field：待比较的字段名
- comp_op：比较方法

多个参考值写在 <ConstValue> 标签列表里。

2.5.3 <Loop> 标签

```
<!-- Loop示例 -->
<Loop name="elementary_stream_loop" length_field="implicit"
length_correction="-4">
  <LoopPresentation>
    <!-- NoLoopHeader和LoopHeader标签二选一 -->
    <NoLoopHeader/>
    <LoopHeader str="节目基本流描述"/>
    <LoopEmpty str="无节目ES信息描述"/>
    <LoopEntry>
      <Prefix str="基本流"/>
    </LoopEntry>
  </LoopPresentation>
  <Body>
    <Field name="stream_type" length="8" encoding="uimsbf">
      <FieldPresentation>
        <Prefix str="流类型"/>
        <Format str="%d"/>
      </FieldPresentation>
    </Field>
  </Body>
</Loop>
```

```

</Field>
<Field name="reserved" length="3" encoding="bslbf"/>
<Field name="elementary_PID" length="13" encoding="uimsbf">
  <FieldPresentation>
    <Prefix str="ES PID"/>
    <Format str="0x%04X (%d)"/>
  </FieldPresentation>
</Field>
<Field name="reserved" length="4" encoding="bslbf"/>
<Field name="ES_info_length" length="12" encoding="uimsbf"/>
<Loop name="ES_info_loop" length_field="ES_info_length">
  <LoopPresentation>
    <NoLoopHeader/>
    <LoopEmpty str="无基本流信息描述符"/>
  </LoopPresentation>
  <Body>
    <Descriptor/>
  </Body>
</Loop>
</Body>
</Loop>

```

<Loop> 标签包含可选的 <LoopPresentation> 标签和强制的 <Body> 标签。<Body> 标签描述循环体的结构。当循环为描述符循环时，只需定义一个描述占位符（<Descriptor>）。

2.5.3.1 <LoopPresentation> 标签

```

<LoopPresentation>
  <NoLoopHeader/>
  <LoopEmpty str="无节目ES信息描述"/>
  <LoopEntry>
    <Prefix str="基本流"/>
  </LoopEntry>
</LoopPresentation>

```

<LoopPresentation> 标签定义循环头标题、循环体标题，以及当循环为空时的标题。也可以设置循环为无头循环（无标题），这样可以减少一层循环展示深度。

<NoLoopHeader> 声明无循环头 <LoopEmpty> 同 <Label> 标签，声明当循环为空时显示的内容。

<LoopEntry> 声明每个循环体的标题。它可以包含以下之一的 <Label> 标签：

- <Fixed>：所有循环体都使用相同的标题。
- <Prefix>：每个循环体使用“前缀+序号”作为标题。序号从1开始。

2.5.3.2 <Body> 标签

<Body> 标签定义循环体结构，可以由一个或以上的 <Syntax> 标签构成，也可以有一个 <Descriptor> 占位符标签构成。当循环为描述符循环时，不再额外添加循环体标题节点。

2.5.4 <Descriptor> 标签

<Descriptor> 标签仅用在循环体定义中，作为任意描述符的占位符。描述符的具体结构由 <DescriptorTemplate> 模板定义。