

XML 与 Vocaloid 3 文件格式

Tom Li

2012 年 8 月 1 日

目录

1 XML 简介	2
1.1 产生及作用	2
1.2 那还有什么用？	2
1.3 XML 的语法	2
2 Vocaloid 3 所使用的 XML	4
2.1 起始区域	4
2.2 歌手设置	4
2.3 调音台设置	5
2.4 主音轨信息	7
2.5 歌手音轨信息	8
2.6 seTrack	11
2.7 karaokeTrack	11
2.8 结束部分	11

版权声明

本文章采用知识共享 [Attribution-NonCommercial-ShareAlike 2.5 China Mainland](#) 许可协议进行许可。您必需遵守该许可协议，否则将可能受到严厉的刑事及民事制裁，并将在法律许可的范围内受到最大可能的起诉。

1 XML 简介

1.1 产生及作用

XML，全称为可扩展标记语言（eXtensible Markup Language），是 1988 年 2 月发布的 W3C¹标准。通过此种标记，计算机之间可以处理包含各种信息的文章等。

但值得注意的是，XML 并没有规定程序应该如何处理和展现这些信息，它起到的作用仅仅是携带信息。也就是说，我可以把文字、数据、算式等一切数据按照 XML 的格式保存在 XML 文件中。至于如何展现信息，则是程序自己需要处理的。这也是 XML 被成为**可扩展**标记语言的原因。

1.2 那还有什么用？

既然 XML 仅仅携带了信息，但并没有指出该怎么处理这些信息，那到底有何意义呢？我把它存到一个普通的文本中也可以实现携带信息啊？

考虑以下的 XML 文件：

```
<荣与耻>
  <条目>以热爱祖国为荣，以危害祖国为耻吗？</条目>
  <条目>以服务人民为荣，以背离人民为耻吗？</条目>
</荣与耻>
```

我们可以看到，XML 文件里面仅仅携带了一些信息。至于这些信息会在屏幕上以红色背景显示出来，以语音形式播放出来，还是出现在 CCTV 上，就完全需要靠程序自己决定了。XML 携带信息，而描述信息则是程序的事。这也就意味着，你只要按照这个程序 XML 的格式，就可以读出信息来，以不同的角度描述这些信息。

1.3 XML 的语法

就拿一份简单的 XML 为例吧：

```
<?xml version="1.0"?>
<观点
```

¹World Wide Web Consortium，万维网联盟。由互联网之父创立，旨在制定和规范互联网的行业标准的组织。

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="观点.xsd">
  <李宗苗><![CDATA[车辆年检，195元/辆，全深圳300万辆，年收5.85
亿元。]]></李宗苗>
```

```
  <杨佩昌><![CDATA[在德国，汽车、公司都无需年检。我问德国人：
汽车不年检，坏了怎么办？答：自己修车。问：企业不年检，公司倒闭如
潮怎么办？德国人答：你看德国是这样的吗？我问：德国政府为何不强制
年检？德国人反问：谁给了政府这个权力？如果政府对年检感兴趣，说明
这种事对它有好处。]]></杨佩昌>
```

```
  <张思之><![CDATA[曾以为一国家只要有好的外交或者强大的国防
就有面子，后来才明白其实人权才是一个国家最大的面子。...一个政权
如果关起门来都不能维护自己本国普通公民的权利，它有什么资格在国际
舞台上维护这个国家的权利，它的合法性就会遭到质疑。]]></张思之>
</观点>
```

这一行叫做 **XML 序言**，它的作用是表明这是一个 1.0 版的 XML 文件。²

而后面几行就特别了，这通常是 XML Schema³。那么什么是 XML Schema 呢？我们之前说过，XML 仅仅携带了信息，而且可以携带任何信息。但是，一个程序不可能可以去读出任意的 XML 文件。每个程序必然有自己定义的 XML。而 XML Schema，就是关于这个程序 XML 文件的格式（比如应该有什么元素啊，元素叫什么名字等等）。只要阅读这份文件，那么这些花里胡哨的 XML 标签到底要按照什么格式填写，就一目了然了。

接下来 XML 就算正式开始了，每一个元素都以<元素>的形式开始，</元素>的形式结束。元素里可以镶嵌更多元素。但说了这么半天，我们发现在上面的 XML 中，“观点”是第一个出现的元素，甚至出现在 XML Schema 之前。这叫做**根元素**，也就是最基本的元素。那它的名字遵循什么标准？Aha，也是 XML Schema。

另外两点：如果你的 XML 保存了大量的信息，甚至有方括号等特殊符号，就会与 XML 本身混淆，只需要在元素中间使用<![CDATA[你的八卦信息]]>的形式，XML 就会认为这只是文字，里面的信息不影响 XML；假设你要在 XML 文件实在让人无法理解，可以直接在 XML 写一些说明

²由于 XML 的特点，所以从 W3C 制定后就基本没做修改了。

³就是“XML 纲要”的意思，但是不知为何，业界不建议对它进行翻译。

之类的东西，只要按照<!-- 说明 --> 的形式写，程序就会很自然的把这些内容忽略掉。

2 Vocaloid 3 所使用的 XML

从 Vocaloid 3 开始，所有的 Vocaloid 歌曲信息也都保存在 XML 文件里了。所谓的.vsqx 文件，其实就是一 XML 文件，是可以直接用文本编辑器打开编辑的。Tom Li 分析文件的时候，最先想看的就是它的 XML Schema，但是 Yamaha 却没有公布，而贴出了一个 404 未找到的无效链接，这意味着我们只能进行黑箱分析了。好在 Vocaloid 所用的 XML 元素的名称还挺容易理解的，对照 Vocaloid，就可以大概知道其中的含义。然而最后，当我在 Vocaloid 的根目录上找到了日文版的 XML Schema 后泪流满面，黑箱分析是白费力气。但是本文已成，所以将大修，重写一文。

2.1 起始区域

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<vsq3 xmlns="http://www.yamaha.co.jp/vocaloid/schema/vsq3/"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.yamaha.co.jp/vocaloid/schema/vsq3/ vsq3.xsd">
  <vender><![CDATA[Yamaha corporation]]></vender>
  <version><![CDATA[3.0.0.11]]></version>
```

XML 的最前面实在是没有什么特别的，我们可以看到 XML 序言，XML Schema 和根元素的名称。根元素叫vsq3，大概是 Vocaloid Queue 3 的意思吧。接下来，我们看到了vender 和version 两个元素，表示这是 Yamaha 公司出品，版本是 3.0.0.11，应该能和 Vocaloid 对应上。

2.2 歌手设置

接下来，一些有趣的元素就出现了。

```
<vVoiceTable>
  <vVoice>
    <vBS>4</vBS>
```

```

    <vPC>0</vPC>
    <compID><![CDATA[BETDB8W6KWZPYEB9]]></compID>
    <vVoiceName><![CDATA[Tianyi_CHN]]></vVoiceName>
    <vVoiceParam>
        <bre>0</bre>
        <bri>0</bri>
        <cle>0</cle>
        <gen>0</gen>
        <ope>0</ope>
    </vVoiceParam>
</vVoice>
</vVoiceTable>

```

首先，vVoiceTable 元素表明里面是 Vocaloid 语音库的数据。在里面，我们看到了一个vVoice 元素，这表示一个歌手。我们可以添加多个vVoice 到vVoiceTable 里面。在往里看，我们看到vBS，这个就比较有意思了，代表语音库的语言。根据日文文档，0 是日语，1 是英文，并且允许后来者加入，显然中文是 4 了，不知道 3 是保留给谁的。还有vPC，看看日文：虚拟程序的变化（其实我不懂日文），应该是指虚拟歌手的编号，不能重复，0 到 127 之间。

不过接下来的compID 和vVoiceName 就好说了，comID 里面是一串咒语一样的字符串，经过验证，发现是洛天依语音库文件夹的名字，便可以知道BETDB8W6KWZPYEB9 肯定是洛天依语音库的编号了。vVoiceName 则是在歌手设置里面给歌手起的名字，本人的文件中是默认的Tianyi_CHN，与实际情况相符。

接下来又见vVoiceParam 嵌套着多个元素。编过程序并有一定水平的人都知道，Param 是“参数”的简写，也就是说里面的内容肯定和歌手的参数有关了。而通过观察，我们不难发现这就是歌手设置里面的四参数，即Breathiness, Brightness, Clearness 和Gender Factor 的缩写。而那迷一样的ope 就不知为何物了，其实应该是Opening，没有错，但就是找不到 Vocaloid 对应的选项。

2.3 调音台设置

下面的乍一看就知道是调音台的配置：

```
<mixer>
  <masterUnit>
    <outDev>0</outDev>
    <retLevel>0</retLevel>
    <vol>0</vol>
  </masterUnit>
  <vsUnit>
    <vsTrackNo>0</vsTrackNo>
    <inGain>0</inGain>
    <sendLevel>-898</sendLevel>
    <sendEnable>0</sendEnable>
    <mute>0</mute>
    <solo>0</solo>
    <pan>64</pan>
    <vol>0</vol>
  </vsUnit>
  <seUnit>
    <inGain>0</inGain>
    <sendLevel>-898</sendLevel>
    <sendEnable>0</sendEnable>
    <mute>0</mute>
    <solo>0</solo>
    <pan>64</pan>
    <vol>0</vol>
  </seUnit>
  <karaokeUnit>
    <inGain>0</inGain>
    <mute>0</mute>
    <solo>0</solo>
    <vol>-129</vol>
  </karaokeUnit>
</mixer>
```

在Mixer 中,最先出现的是masterUnit,就是 Vocaloid 3 中“调音台”最右侧的总音效控制了。OutDev 不知是何物,retLevel 指的就是调音台中那个return 旋钮的数值了。但这个数值是扩大 10 倍后的整数。假如你的return 是 52.5,那么这里一定是 525。vol 自然就是下面的音量滑块的数值了,同样是扩大 10 倍以后的值。inGain 就是调音台中Gain 旋钮的值了。

接下来的vsUnit,就是歌手所在的音轨了(我猜 vs 可能代表Virtual Singer 吧),由调音台的第一栏所控制。vsTrackNo 就是音轨编号了。我猜,既然第一条音轨的编号为 0,第二条就为 1 了,以此类推。sendLevel 就是调音台中send 旋钮的值,不过扩大了 10 倍。sendEnable 则是send 旁边的那个小开关,编过程序的肯定知道这是一个布尔值——如果打开了,就是 1,否则就是 0。mute 对应着静音用的M 按钮,solo 自然对应着s 按钮,都是布尔值,如果打开了,就是 1,否则就是 0。pan 是左右声道平衡,64 代表平衡,就是 0 的意思,如果你设置偏向右声道 6 个单位,数值就变 70 了;往左正好相反。最后的Vol 同样是下面的音量滑块的数值了,同样是扩大 10 倍以后的值。

而seUnit 则是第二栏,karaokeUnit 则是第三栏,参数没有变化,在前面都已经解释完了。

2.4 主音轨信息

```
<masterTrack>
  <seqName><![CDATA[Untitled0]]></seqName>
  <comment><![CDATA[New VSQ File]]></comment>
  <resolution>480</resolution>
  <preMeasure>5</preMeasure>
  <timeSig>
    <posMes>0</posMes>
    <nume>4</nume>
    <denomi>4</denomi>
  </timeSig>
  <tempo>
    <posTick>0</posTick>
    <bpm>12000</bpm>
```

```

    </tempo>
</masterTrack>

```

masterTrack 这个元素名称，很显然是主控，也就是整体歌曲的设置。元素中，seqName 是一个字符串，代表序列名称。类似的，comment 就代表序列的注释了。resolution 是时间分辨率⁴，目前 Vocaloid 3 把它固定成了 480。preMeasure 了，代表默认前置小节，也就是在歌曲开始前有几个空白小节。

再来看timeSig 元素，posMes 代表拍号所在的小节，nume 和denomi 代表每拍时值。tempo 则是和速率有关的参数，posTick 代表速率记号所处位置（音位）；bmp 代表进行速率。

2.5 歌手音轨信息

这里就是 Vocaloid 谜一样的地方了，为了避免太多的复杂性，就以一个歌手为洛天依，只有一拍 C3 调的歌词“a”为例吧。

```

<vsTrack>
  <vsTrackNo>0</vsTrackNo>
  <trackName><![CDATA[Track]]></trackName>
  <comment><![CDATA[Track]]></comment>
  <musicalPart>
    <posTick>9600</posTick>
    <playTime>1920</playTime>
    <partName><![CDATA[NewPart]]></partName>
    <comment><![CDATA[New Musical Part]]></comment>
    <stylePlugin>
      <stylePluginID><![CDATA[ACA9C502-A04B-42b5-B2EB-5CEA36D16FCE]]></stylePluginID>
      <stylePluginName><![CDATA[VOCALOID2 Compatible Style]]></stylePluginName>
      <version><![CDATA[3.0.0.1]]></version>
    </stylePlugin>
    <partStyle>
      <attr id="accent">50</attr>

```

⁴日文“分解能”，“分解能”即分辨率，XML 中的 Resolution 也代表分辨率。故将其翻译为“时间分辨率”，不是没有根据的。

```

    <attr id="bendDep">8</attr>
    <attr id="bendLen">0</attr>
    <attr id="decay">50</attr>
    <attr id="fallPort">0</attr>
    <attr id="opening">127</attr>
    <attr id="risePort">0</attr>
</partStyle>
<singer>
    <posTick>0</posTick>
    <vBS>4</vBS>
    <vPC>0</vPC>
</singer>
<note>
    <posTick>0</posTick>
    <durTick>480</durTick>
    <noteNum>60</noteNum>
    <velocity>64</velocity>
    <lyric><![CDATA[a]]</lyric>
    <phnms><![CDATA[a]]</phnms>
    <noteStyle>
        <attr id="accent">50</attr>
        <attr id="bendDep">8</attr>
        <attr id="bendLen">0</attr>
        <attr id="decay">50</attr>
        <attr id="fallPort">0</attr>
        <attr id="opening">127</attr>
        <attr id="risePort">0</attr>
        <attr id="vibLen">66</attr>
        <attr id="vibType">1</attr>
        <seqAttr id="vibDep">
            <elem>
                <posNrm>22391</posNrm>
                <elv>64</elv>
            </elem>
        </seqAttr>
    </noteStyle>
</note>

```

```

        </elem>
    </seqAttr>
    <seqAttr id="vibRate">
        <elem>
            <posNrm>22391</posNrm>
            <elv>50</elv>
        </elem>
    </seqAttr>
</noteStyle>
</note>
</musicalPart>
</vsTrack>

```

在`vsTrack` 元素中，`vsTrackNo` 代表歌手音轨的编号，在这里只有一个音轨，所以是 0，`TrackName` 和`comment` 表示音轨注释，两者都是字符串，而且是在 Vocaloid 的属性中编辑的。

在接下里的子元素`musicalPart` 中，第一个元素`posTick` 代表其所处的位置（包括前面的空白小节部分），数值规定同`playTime`。`playTime` 就神奇了，这个元素代表该音轨的播放时间。关键是，其实通过试验 Tom Li 发现，这个数值可以任意设置：如果你的歌的长度只有 1-2 一个区间，那么长度将是 1920，如果不足一个区间，比如 20，就代表一个区段的 $\frac{1}{96}$ 了，以此类推。`stylePlugin` 则是有关该音轨样式的插件信息，和你安装的插件有关。这里的`stylePluginID` 和`stylePluginName` 分别是字符串，表示插件的标号和名称。默认自带的这个插件，从名称看可能和 Vocaloid 2 的兼容有关系。

`partStyle` 则是你演唱风格中的整体设置。从名字就可以看出来，`accent` 和`decay` 是重音和衰减的百分比；`bendDep` 和`bendLen` 是高音曲折幅度和高音曲折时长；`fallPort` 和`risePort` 则是布尔值，决定是否有下降滑音和上升滑音。对于`opening`，应该是默认的开合度，但似乎不能直接设置。

接下来的元素是`singer`，是歌手的一些参数，但是 Tom Li 并没有找出其中的含义，所以直接看`note` 吧。这一部分内容就是和唱的内容有很大关系的了。`posTick`, `durTick` 的意思分别这个音符从何时开始，长度是多少，计时方法和`playTime` 一致。`noteNum` 和`velocity`，我随便猜了猜，没

有做验证，应该是音符的编号的速度。lyric 和phnms，是这个音符的歌词和音标⁵。看到noteStyle 的时候，那些乱七八糟的参数就又来了。前面的七个参数是 Tom Li 之前说过的，后面的以vib 打头的几个参数，想想英文就知道这肯定是控制颤音的。vibLen 和vibType 是颤音长度和颤音类型。而vibDip 和vibRate 是振幅和周期，不过其实这是两个列表，我们看到里面只有一项，当设置曲折的振幅和周期的时候，列表就会有多项了。这两个列表的格式一致，那就一起说喽，posNrm 还是计时单位，不过这个时间单位和前面的完全不同，elv 就代表数值了，可以是振幅和周期。

2.6 seTrack

```
<seTrack>
</seTrack>
```

暂时不分析这条音轨。

2.7 karaokeTrack

```
<karaokeTrack>
</karaokeTrack>
```

暂时不分析这条音轨。

2.8 结束部分

```
<aux>
  <auxID><![CDATA[AUX_VST_HOST_CHUNK_INFO]]></auxID>
  <content><![CDATA[VlNDSwAAAAADAAAAAAAAAAAAAAAAAAAAAAAAAAAA=]]></content>
</aux>
</vsq3>
```

这里是一些常规信息，别动它了。

⁵别忘了 Vocaloid 是可以单独指定另一个音标给一个读音的