

TC260-PG-202510A

网络安全标准实践指南

——人工智能生成合成内容标识方法 文件
元数据隐式标识 音频文件

(V1.0-202508)



全国网络安全标准化技术委员会秘书处

2025 年 08 月

本文档可从以下网址获得：

www.tc260.org.cn/



全国网络安全标准化技术委员会
National Technical Committee 260 on Cybersecurity of SAC



前 言

《网络安全标准实践指南》（以下简称《实践指南》）是全国网络安全标准化技术委员会（以下简称“网安标委”）秘书处组织制定和发布的标准相关技术文件，旨在围绕网络安全法律法规政策、标准、网络安全热点和事件等主题，宣传网络安全相关标准及知识，提供标准化实践指引。

本文件起草单位：腾讯音乐娱乐（深圳）有限公司、中国电子技术标准化研究院、国家计算机网络应急技术处理协调中心、浙江大学、北京抖音信息服务有限公司、北京快手科技有限公司、国家计算机网络应急技术处理协调中心山西分中心、中国移动通信集团有限公司、中央网信办数据与技术保障中心、阿里云计算有限公司、北京金山办公软件股份有限公司。

本文件主要起草人：陈洲旋、李茹、李伟峰、郝春亮、许晓耕、孙硕、马文涛、董琳、王志伟、张立尧、落红卫、谷晨、贺凯、杜蕾、王秉政、卢立、倪王泽、贺敏、孙培尧、费凡芮、吴巍、杨敏、徐嵩、张秋芬、马丽萌、孙勇、张树玲、安红云。



声 明

本《实践指南》版权属于网安标委秘书处，未经秘书处书面授权，不得以任何方式抄袭、翻译《实践指南》的任何部分。凡转载或引用本《实践指南》的观点、数据，请注明“来源：全国网络安全标准化技术委员会秘书处”。



全国网络安全标准化技术委员会
National Technical Committee 260 on Cybersecurity of SAC



摘 要

为落实《人工智能生成合成内容标识办法》，根据强制性国家标准GB 45438—2025《网络安全技术 人工智能生成合成内容标识方法》的要求，本文件提供了人工智能生成合成的音频文件元数据隐式标识方法，包括针对不同文件格式的具体方案，指导人工智能生成合成内容服务提供者和网络信息内容传播服务提供者开展人工智能生成合成的音频文件元数据隐式标识活动。





目 录

1 范围	1
2 规范性引用文件	1
3 术语和定义	1
4 缩略语	2
5 概述	2
5.1 音频文件格式	2
5.2 方法概述	3
6 音频文件的元数据隐式标识方案	3
6.1 原生嵌入方案	3
6.2 XMP 方案	6
附录 A （资料性） WAV 音频文件示例	7
附录 B （资料性） MP3 音频文件示例	8
附录 C （资料性） OGG 音频文件示例	9
附录 D （资料性） FLAC 音频文件示例	10
附录 E （资料性） M4A 音频文件示例	11
附录 F （资料性） XMP 数据的构造示例	13
附录 G （资料性） XMP 写入实现参考示例	14
参考文献	16





1 范围

本文件给出了人工智能生成合成音频内容的文件元数据隐式标识方法，包括针对不同音频文件格式的具体方案。

本文件适用于指导人工智能生成合成音频内容服务提供者和网络信息音频内容传播服务提供者开展文件元数据隐式标识活动。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB 45438—2025 网络安全技术 人工智能生成合成内容标识方法

3 术语和定义

下列术语和定义适用于本文件。

3.1 人工智能生成合成音频内容 **audio content generated by artificial intelligence**

利用人工智能技术生成、合成的音频信息。

3.2 人工智能生成合成音频内容服务提供者 **artificial intelligence audio content generation service provider**

生成合成音频服务提供者

利用人工智能技术（包括通过提供可编程接口等方式）向公众提



供生成合成音频服务的组织或个人。

3.3 网络信息音频内容传播服务提供者 internet information audio content propagation service provider

音频内容传播服务提供者

提供网络信息音频内容传播服务的网络信息服务提供者。

4 缩略语

下列缩略语适用于本文件。

AVS: 数字音视频编解码技术标准 (Advanced Audio-Video Coding/Decoding Standard)

AAC: 高级音频编码 (Advanced Audio Coding)

FLAC: 无损音频压缩编码 (Free Lossless Audio Codec)

MP3: 动态影像专家压缩标准频层面 3 (Moving Picture Experts Group Audio Layer III)

M4A: MPEG-4 音频标准文件的扩展名 (MPEG-4 Audio)

OGG: 音频压缩格式 (OGG Vorbis)

OPUS: Opus 交互音频编码 (Opus Interactive Audio Codec)

RIFF: 资源交换文件格式 (Resource Interchange File Format)

WAV: 波形音频文件格式 (Waveform Audio File Format)

XMP: 可扩展元数据平台 (Extensible Metadata Platform)

5 概述

5.1 音频文件格式



音频文件格式一般包括：

- a) 基于 RIFF 规范的音频文件，如 WAV；
- b) 基于 MP3 规范的音频文件，如 mp3；
- c) 基于 OGG 规范的音频文件，如 Vorbis、OPUS；
- d) 基于 FLAC 规范的音频文件，如 flac；
- e) 基于 M4A 规范的音频文件，如 AVS、AAC。

5.2 方法概述

针对本文件 5.1 所列出的各类格式的音频文件，本文件给出了通过原生嵌入实现的元数据隐式标识方案；针对其他格式的音频文件，同时给出了通过 XMP 规范实现的元数据隐式标识方案。

注：原生嵌入指利用特定文件格式预先定义的存储结构与访问接口将元数据写入文件内部预留区域的行为。

6 音频文件的元数据隐式标识方案

6.1 原生嵌入方案

6.1.1 WAV 音频文件

方案如下：

- a) 存储位置 RIFF 文件中的 Subchunk 中，其中可定义 Subchunk 为 AIGC；
- b) 写入方式为：
 - 1) 在 RIFF 文件中插入 Subchunk 块，定义 Subchunk ID 为 AIGC；



- 2) 在 1) 中 ID 为 AIGC 的 Subchunk 块的 Subchunk Data 中,
写入 GB 45438—2025 附录 E b) 所定义的字符串:
`{"Label":"value1","ContentProducer":"value2","ProduceID
":"value3","ReservedCode1":"value4","ContentPropagator":
"value5","PropagateID":"value6","ReservedCode2":"value7
"}`。

WAV 音频文件示例参见附录 A。

6.1.2 MP3 音频文件

方案如下:

- a) 存储位置为 ID3v2 的 TXXX 帧, 即 User defined text information frame;
- b) 写入方式为:

- 1) 在 ID3v2 中, 定义一个标签帧为 TXXX_AIGC; 具体来说,
是 TXXX 帧, 其描述 (Description) 写入 AIGC;
- 2) 在标签帧为 AIGC 的帧内容中写入 GB 45438—2025 附录
E b) 所定义的字符串:
`{"Label":"value1","ContentProducer":"value2","ProduceID
":"value3","ReservedCode1":"value4","ContentPropagator":
"value5","PropagateID":"value6","ReservedCode2":"value7
"}`。

MP3 音频文件示例参见附录 B。



6.1.3 OGG 音频文件

对于 OGG 规范的 Vorbis 音频文件、OPUS 音频文件，方案如下：

- a) 存储位置为 VORBIS_COMMENT 块；
- b) 写入方式为：

在 VORBIS_COMMENT 块中，写入 key: AIGC，其 value 值可写入 GB 45438—2025 附录 E b) 所定义的字符串：

```
{"Label":"value1","ContentProducer":"value2","ProduceID":"value3","ReservedCode1":"value4","ContentPropagator":"value5","PropagateID":"value6","ReservedCode2":"value7"}。
```

OGG 音频文件示例参见附录 C。

6.1.4 FLAC 音频文件

方案如下：

- a) 存储位置为 VORBIS_COMMENT 块；
- b) 写入方式为：

在 VORBIS_COMMENT 块中，写入 key: AIGC，其 value 值写入 GB 45438—2025 附录 E b) 所定义的字符串：

```
{"Label":"value1","ContentProducer":"value2","ProduceID":"value3","ReservedCode1":"value4","ContentPropagator":"value5","PropagateID":"value6","ReservedCode2":"value7"}。
```

FLAC 音频文件示例参见附录 D。

6.1.5 M4A 音频文件



对于 M4A 规范的 AVS 音频文件、AAC 音频文件，方案如下：

a) 存储位置为 moov.udta.meta.;

b) 写入方式为：

在 moov.udta.meta.keys 中写入 key: AIGC;

在 moov.udta.meta.ilst 中写入 value:

```
{"Label":"value1","ContentProducer":"value2","ProduceID":"value3","ReservedCode1":"value4","ContentPropagator":"value5","PropagateID":"value6","ReservedCode2":"value7"}。
```

M4A 音频文件示例参见附录 E。

6.2 XMP 方案

对于其他音频格式，XMP 方案如下：

a) RDF 中新增自定义命名空间 TC260，URI 为 <http://www.tc260.org.cn/ns/AIGC/1.0/>。该命名空间已有其他内容时，应按属性更新，不应整体覆盖；

b) 在 TC260 空间下 AIGC 键值中填入 GB 45438—2025 附录 E 规定字符串，且按照下文实现序列化：

```
<TC260:AIGC>{"Label":"value1","ContentProducer":"value2","ProduceID":"value3","ReservedCode1":"value4","ContentPropagator":"value5","PropagateID":"value6","ReservedCode2":"value7"}</TC260:AIGC>。
```

XMP 方案示例参见附录 F、附录 G。



附录 A

(资料性)

WAV 音频文件示例

以 exiftool 工具开发为例，实现 WAV 音频文件元数据读取的示例。

读取元数据的示例如下：

```
exiftool -a -G1 -v4 ./sample.wav
RIFF 'fmt ' chunk (16 bytes of data):
.....
RIFF 'AIGC' chunk (168 bytes of data):
  Unknown_AIGC =
  {"Label":"value1","ContentProducer":"value2","ProduceID":"value3","ReservedCode1":"value4","ContentPropagator":"value5","PropagateID":"value6","ReservedCode2":"value7"}
  - Tag 'AIGC' (168 bytes):
    002c: 7b 22 4c 61 62 65 6c 22 3a 22 76 61 6c 75 65 31 [{"Label":"value1"}
    003c: 22 2c 22 43 6f 6e 74 65 6e 74 50 72 6f 64 75 63 [{"ContentProducer":
    004c: 65 72 22 3a 22 76 61 6c 75 65 32 22 2c 22 50 72 [er":"value2","ProduceID":"value3"}
    005c: 6f 64 75 63 65 49 44 22 3a 22 76 61 6c 75 65 33 [oduceID":"value3"}
    006c: 22 2c 22 52 65 73 65 72 76 65 64 43 6f 64 65 31 [{"ReservedCode1":
    007c: 22 3a 22 76 61 6c 75 65 34 22 2c 22 43 6f 6e 74 [":"value4","ContentPropagator":
    008c: 65 6e 74 50 72 6f 70 61 67 61 74 6f 72 22 3a 22 [entPropagator":
    009c: 76 61 6c 75 65 35 22 2c 22 50 72 6f 70 61 67 61 [value5","PropagateID":"value6"}
    00ac: 74 65 49 44 22 3a 22 76 61 6c 75 65 36 22 2c 22 [teID":"value6","ReservedCode2":
    00bc: 52 65 73 65 72 76 65 64 43 6f 64 65 32 22 3a 22 [ReservedCode2":
    00cc: 76 61 6c 75 65 37 22 7d [value7"}
  .....
```



附录 B

(资料性)

MP3 音频文件示例

以 ffmpeg 工具和 exiftool 工具开发为例，实现 MP3 音频文件元数据写入和读取的示例。

写入元数据的示例如下：

```
ffmpeg -i input.mp3 \
-metadata
AIGC="{\"Label\":\"value1\",\"ContentProducer\":\"value2\",\"ProduceID\":\"value3\",\"ReservedCode1\":\"value4\",
\"ContentPropagator\":\"value5\",\"PropagateID\":\"value6\",\"ReservedCode2\":\"value7\"}"
-c copy sample.mp3
```

读取元数据的示例如下：

```
exiftool -a -G1 -v4 ./sample.mp3
ID3v2.4.0:
.....
| - Tag 'TXXX_AIGC' (175 bytes):
|   0014: 03 41 49 47 43 00 7b 22 4c 61 62 65 6c 22 3a 22 [.AIGC.{"Label":"]
|   0024: 76 61 6c 75 65 31 22 2c 22 43 6f 6e 74 65 6e 74 [value1","Content]
|   0034: 50 72 6f 64 75 63 65 72 22 3a 22 76 61 6c 75 65 [Producer":"value]
|   0044: 32 22 2c 22 50 72 6f 64 75 63 65 49 44 22 3a 22 [2","ProduceID":"]
|   0054: 76 61 6c 75 65 33 22 2c 22 52 65 73 65 72 76 65 [value3","Reserve]
|   0064: 64 43 6f 64 65 31 22 3a 22 76 61 6c 75 65 34 22 [dCode1":"value4"]
|   0074: 2c 22 43 6f 6e 74 65 6e 74 50 72 6f 70 61 67 61 [,"ContentPropaga]
|   0084: 74 6f 72 22 3a 22 76 61 6c 75 65 35 22 2c 22 50 [tor":"value5","P]
|   0094: 72 6f 70 61 67 61 74 65 49 44 22 3a 22 76 61 6c [ropagateID":"val]
|   00a4: 75 65 36 22 2c 22 52 65 73 65 72 76 65 64 43 6f [ue6","ReservedCo]
|   00b4: 64 65 32 22 3a 22 76 61 6c 75 65 37 22 7d 00 [de2":"value7"}.]
|
|.....
```



附录 C

(资料性)

OGG 音频文件示例

以 `ffmpeg` 工具和 `exiftool` 工具开发为例，实现 OGG 音频文件元数据写入和读取的示例。

写入元数据的示例如下：

```
ffmpeg -i input.ogg \  
-metadata  
AIGC="{\"Label\":\"value1\",\"ContentProducer\":\"value2\",\"ProduceID\":\"value3\",\"ReservedCode1\":\"  
value4\",\"ContentPropagator\":\"value5\",\"PropagateID\":\"value6\",\"ReservedCode2\":\"value7\"}\  
-c copy sample.ogg
```

读取元数据的示例如下：

```
exiftool -a -G1 -v4 ./sample.ogg  
.....  
- Tag 'AIGC' (173 bytes):  
0020: 41 49 47 43 3d 7b 22 4c 61 62 65 6c 22 3a 22 76 [AIGC="{\"Label\":\"v]  
0030: 61 6c 75 65 31 22 2c 22 43 6f 6e 74 65 6e 74 50 [alue1\",\"ContentP]  
0040: 72 6f 64 75 63 65 72 22 3a 22 76 61 6c 75 65 32 [roducer\":\"value2]  
0050: 22 2c 22 50 72 6f 64 75 63 65 49 44 22 3a 22 76 [\", \"ProduceID\":\"v]  
0060: 61 6c 75 65 33 22 2c 22 52 65 73 65 72 76 65 64 [alue3\", \"Reserved]  
0070: 43 6f 64 65 31 22 3a 22 76 61 6c 75 65 34 22 2c [Code1\":\"value4\",]  
0080: 22 43 6f 6e 74 65 6e 74 50 72 6f 70 61 67 61 74 [\"ContentPropagat]  
0090: 6f 72 22 3a 22 76 61 6c 75 65 35 22 2c 22 50 72 [or\":\"value5\", \"Pr]  
00a0: 6f 70 61 67 61 74 65 49 44 22 3a 22 76 61 6c 75 [opagateID\":\"valu]  
00b0: 65 36 22 2c 22 52 65 73 65 72 76 65 64 43 6f 64 [e6\", \"ReservedCod]  
00c0: 65 32 22 3a 22 76 61 6c 75 65 37 22 7d [e2\":\"value7\"}]
```



附录 D

(资料性)

FLAC 音频文件示例

以 ffmpeg 工具和 exiftool 工具开发为例, 实现 FLAC 音频文件数据写入和读取的示例。

写入元数据的示例如下:

```
ffmpeg -i input.flac \
-metadata
AIGC="{\"Label\":\"value1\",\"ContentProducer\":\"value2\",\"ProduceID\":\"value3\",\"ReservedCode1\":\"value4\",
\"ContentPropagator\":\"value5\",\"PropagateID\":\"value6\",\"ReservedCode2\":\"value7\"}" \
-c copy sample.flac
```

读取元数据的示例如下:

```
exiftool -a -G1 -v4 ./sample.flac
VorbisComment (SubDirectory) -->
- Tag 0x0004 (223 bytes):
  002e: 0d 00 00 00 4c 61 76 66 35 37 2e 37 31 2e 31 30 [...Lavf57.71.10]
  003e: 30 02 00 00 00 ad 00 00 00 41 49 47 43 3d 7b 22 [0.....AIGC={"
  004e: 4c 61 62 65 6c 22 3a 22 76 61 6c 75 65 31 22 2c [Label":"value1",]
  005e: 22 43 6f 6e 74 65 6e 74 50 72 6f 64 75 63 65 72 [ContentProducer]
  006e: 22 3a 22 76 61 6c 75 65 32 22 2c 22 50 72 6f 64 [":"value2","Prod]
  007e: 75 63 65 49 44 22 3a 22 76 61 6c 75 65 33 22 2c [uceID":"value3",]
  008e: 22 52 65 73 65 72 76 65 64 43 6f 64 65 31 22 3a [ReservedCode1":]
  009e: 22 76 61 6c 75 65 34 22 2c 22 43 6f 6e 74 65 6e [value4","Conten]
  00ae: 74 50 72 6f 70 61 67 61 74 6f 72 22 3a 22 76 61 [tPropagator":"va]
  00be: 6c 75 65 35 22 2c 22 50 72 6f 70 61 67 61 74 65 [lue5","Propagate]
  00ce: 49 44 22 3a 22 76 61 6c 75 65 36 22 2c 22 52 65 [ID":"value6","Re]
  00de: 73 65 72 76 65 64 43 6f 64 65 32 22 3a 22 76 61 [servedCode2":"va]
  00ee: 6c 75 65 37 22 7d 15 00 00 00 65 6e 63 6f 64 65 [lue7"}....encode]
  00fe: 72 3d 4c 61 76 66 35 37 2e 37 31 2e 31 30 30 [r=Lavf57.71.100]
.....
```



附录 E

(资料性)

M4A 音频文件示例

以 ffmpeg 工具和 exiftool 工具开发为例，实现 M4A 音频文件数据写入和读取的示例。

写入元数据的示例如下：

```
ffmpeg -i input.m4a \  
-metadata  
AIGC="{\"Label\":\"value1\",\"ContentProducer\":\"value2\",\"ProduceID\":\"value3\",\"ReservedCode1\":\"  
value4\",\"ContentPropagator\":\"value5\",\"PropagateID\":\"value6\",\"ReservedCode2\":\"value7\"}\  
-movflags use_metadata_tags \  
-c copy sample.m4a
```

读取元数据的示例如下：

```
exiftool -a -G1 -v4 ./sample.m4a  
.....  
||| Keys (SubDirectory) -->  
||| - Tag 'keys' (116 bytes):  
||| 1e7d6: 00 00 00 00 00 00 00 06 00 00 00 13 6d 64 74 61 [.....mdta]  
||| 1e7e6: 6d 61 6a 6f 72 5f 62 72 61 6e 64 00 00 00 15 6d [major_brand....m]  
||| 1e7f6: 64 74 61 6d 69 6e 6f 72 5f 76 65 72 73 69 6f 6e [dtaminor_version]  
||| 1e806: 00 00 00 19 6d 64 74 61 63 6f 6d 70 61 74 69 62 [....mdtacompatib]  
||| 1e816: 6c 65 5f 62 72 61 6e 64 73 00 00 00 0c 6d 64 74 [le_brands....mdt]  
||| 1e826: 61 41 49 47 43 00 00 00 10 6d 64 74 61 69 54 75 [aAIGC....mdtaiTu]  
||| 1e836: 6e 53 4d 50 42 00 00 00 0f 6d 64 74 61 65 6e 63 [nSMPB....mdtaenc]  
||| 1e846: 6f 64 65 72 [oder]  
||| + [Keys directory]  
||| Added ItemList Tag 1.1 = (mdta) major_brand  
||| Added ItemList Tag 1.2 = (mdta) minor_version  
||| Added ItemList Tag 1.3 = (mdta) compatible brands  
||| Added ItemList Tag 1.4 = (mdta) AIGC (Unknown)  
||| Added ItemList Tag 1.5 = (mdta) iTunSMPB (Unknown)  
||| Added ItemList Tag 1.6 = (mdta) encoder  
||| ItemList (SubDirectory) -->  
||| - Tag 'ilst' (457 bytes):  
||| 1e852: 00 00 00 1c 00 00 00 01 00 00 00 14 64 61 74 61 [.....data]  
||| 1e862: 00 00 00 01 00 00 00 00 6d 70 34 32 00 00 00 19 [.....mp42....]  
||| 1e872: 00 00 00 02 00 00 00 11 64 61 74 61 00 00 00 01 [.....data....]  
||| 1e882: 00 00 00 00 30 00 00 00 24 00 00 00 03 00 00 00 [....0...$.....]  
||| 1e892: 1c 64 61 74 61 00 00 00 01 00 00 00 00 4d 34 41 [.data.....M4A]  
||| 1e8a2: 20 6d 70 34 32 69 73 6f 6d 00 00 00 c0 00 00 00 [ mp42isom.....]  
||| 1e8b2: 04 00 00 00 b8 64 61 74 61 00 00 00 01 00 00 00 [.....data.....]
```




```
1e8c2: 00 7b 22 4c 61 62 65 6c 22 3a 22 76 61 6c 75 65 [."Label": "value]
1e8d2: 31 22 2c 22 43 6f 6e 74 65 6e 74 50 72 6f 64 75 [1, "ContentProdu
1e8e2: 63 65 72 22 3a 22 76 61 6c 75 65 32 22 2c 22 50 [cer": "value2", "P
1e8f2: 72 6f 64 75 63 65 49 44 22 3a 22 76 61 6c 75 65 [roduceID": "value
1e902: 33 22 2c 22 52 65 73 65 72 76 65 64 43 6f 64 65 [3, "ReservedCode
1e912: 31 22 3a 22 76 61 6c 75 65 34 22 2c 22 43 6f 6e [1": "value4", "Con
1e922: 74 65 6e 74 50 72 6f 70 61 67 61 74 6f 72 22 3a [tentPropagator":]
1e932: 22 76 61 6c 75 65 35 22 2c 22 50 72 6f 70 61 67 [ "value5", "Propag
1e942: 61 74 65 49 44 22 3a 22 76 61 6c 75 65 36 22 2c [ateID": "value6",]
1e952: 22 52 65 73 65 72 76 65 64 43 6f 64 65 32 22 3a [ "ReservedCode2":]
1e962: 22 76 61 6c 75 65 37 22 7d 00 00 00 8c 00 00 00 [ "value7"}.....]
1e972: 05 00 00 00 84 64 61 74 61 00 00 00 01 00 00 00 [.....data.....]
```





附录 F

(资料性)

XMP 数据的构造示例

以新定义命名空间 <http://www.tc260.org.cn/ns/AIGC/1.0/> 为例，构造 XMP 数据的示例如下：

```
<x:xmpmeta xmlns:x="adobe:ns:meta/">
  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:tc260="http://www.tc260.org.cn/ns/AIGC/1.0/">
    <rdf:Description rdf:about="">

      <tc260:AIGC>{"Label":"value1","ContentProducer":"value2","ProduceID":"value3","ReservedCode1":"value4","ContentPropagator":"value5","PropagateID":"value6","ReservedCode2":"value7"}
    </tc260:AIGC>
    </rdf:Description>
  </rdf:RDF>
</x:xmpmeta>
```





附录 G

(资料性)

XMP 写入实现参考示例

以 python 开发实现为例，示例如下：

a) 安装 `exempi`，示例如下：

```
pip install python-xmp-toolkit  
brew install exempi
```

b) 写入 XMP 元数据，示例如下：

```
# tc260.py  
import sys  
from libxmp import XMPMeta, XMPFiles  
  
if __name__ == '__main__':  
    if len(sys.argv) < 2:  
        print('no argument')  
        sys.exit()  
    input_file = sys.argv[1]  
  
    # XMP 数据  
    xmp_data = """<x:xmpmeta xmlns:x="adobe:ns:meta/">  
        <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:tc260="http://www.tc260.org.cn/ns/AIGC/1.0/">  
            <rdf:Deion rdf:about="">  
  
            <tc260:AIGC>{"Label":"value1","ContentProducer":"value2","ProduceID":"value3","ReservedCode1":"value4","ContentPropagator":"value5","PropagateID":"value6","ReservedCode2":"value7"}  
            </tc260:AIGC>  
            </rdf:Description>  
        </rdf:RDF>  
    </x:xmpmeta>"""  
  
    xmp = XMPMeta()  
    xmp.parse_from_str(xmp_data)  
    xmpfile = XMPFiles(file_path=input_file, open_forupdate=True)  
    xmpfile.put_xmp(xmp)  
    xmpfile.close_file()
```

c) 调用，示例如下：

```
python tc260.py example.wmv
```



d) 以使用 exiftool 工具为例，提取 XMP 元数据的示例如下：

```
exiftool -xmp:aigc example.wmv
```





参考文献

[1] Windows 官网，资源交换文件格式（RIFF）：

<https://learn.microsoft.com/zh-cn/windows/win32/xaudio2/resource-interchange-file-format--riff-?redirectedfrom=MSDN>

[2] ISO/IEC 13818-3:1998: Information technology — Generic coding of moving pictures and associated audio information — Part 3: Audio

[3] Ogg Vorbis I format specification: comment field and header specification: <https://xiph.org/vorbis/doc/v-comment.html>

[4] Free Lossless Audio Codec (FLAC) RFC: 9639,
<https://www.rfc-editor.org/rfc/rfc9639.txt>

[5] ISO/IEC 14496-14:2020: Information technology — Coding of audio-visual objects Part 14: MP4 file format

