Audible Magic Toolkit User's Guide

Version 38.11d

AMToolkit UG
rev. 38-11.1.2019

Contents

DISCLAIMER

The information provided herein is intended as a reference to the use of Audible Magic's products and services only and is not intended to be a guidance for copyright compliance on a given website, platform, application or service. Please consult an appropriate legal representative for guidance regarding compliance procedures and policies as it relates to your specific use case.

The Audible Magic Toolkit contains two programs that are able to determine the presence of audio and video material that has previously been registered with Audible Magic.

A program named *identify* accepts an input media file and produces JSON output. The JSON includes the list of any matches that were made between the media file and entries in Audible Magic's databases. The usage details of *identify* are given below.

The toolkit also includes the program *analyze*. *Analyze* is for customers who need to separate the analysis of the media file from the process of communicating with our servers. It is described later, but most customers do not need to be concerned with *analyze.*

Installation

For complete installation instructions, please refer to the *AM ToolKit QuickStart Guide*.

The Audible Magic Toolkit is delivered as a gzipped tarball or a .zip file, depending upon the operating system, with a name like AudibleMagicToolkit_*x*_linux64.tgz or AudibleMagicToolkit_*x*_macOS.tgz or AudibleMagicToolkit_*x*_win64.zip. The *x* in these names is substituted with the release version, such as 38.11.

Unpacking that file produces the following top-level items:

- **AMToolkit_Users_Guide.pdf** - this document.
- **ReleaseNotes.txt –** a list of bug fixes and feature additions in this and previous releases.
- **bin** - a directory containing the *identify* and *analyze* command-line programs and associated binaries and libraries. You will need to execute the command line from within that directory or point your library and bin paths to find it.

Warning: Avoid unpacking a Linux or macOS .tgz file on Windows, as doing so breaks symbolic links within the release package.

Installation on Mac OS

You should set the environment variable `DYLD_LIBRARY_PATH` on the Mac to include the current working directory (i.e., "."). For example, in the bash shell, this can be done by typing:

```
export DYLD_LIBRARY_PATH=.:$DYLD_LIBRARY_PATH
```

Otherwise, you are likely to get a run-time error message saying "Library not loaded."

Configuration Files

You must contact Client Services at Audible Magic to receive a configuration file that is designed for your specific needs. Both programs require a configuration file in order to function.

identify

Usage

The *identify* command-line program is run like this:

identify -c *configFile* -i *mediaFile* -e *unknownAssetID* -o *outputFile*

- *configFile* is the configuration file supplied to you by Audible Magic.
- *mediaFile* is the media file you wish to identify.
- *unknownAssetID* is a unique customer-determined string that identifies the media file in the customer's systems. The "unknown" means that this media file is the one to be identified.
- *outputFile* is the JSON file containing the results of the identification. If omitted, the JSON is written to stdout.

Example

Say the soundtrack of a media file named 82345.mp4 includes the beginning of the song *Feels Like Summer* by Weezer, starting at 60 seconds from the beginning of the file. You would invoke *identify* like this:

```
identify -c my.config -i 82345.mp4 -e 82345 -o 82345.json
```

When the process completes, the JSON output file 82345.json will look something like the following, indicating a match was found, the extent of the match, the metadata associated with the match, and what action to take as a result. (A detailed description of the JSON fields is given later.)


```
{

  "statusCode": 2006,
  "statusDescription": "Status 2006 (2006) Match found",
  "unknownAssetID": "82345",
  "matches": [
        {
              "matchType": "audio",
              "amItemID": "4514a672110f4a4eb96e507b33c592fd_RGUa00",
              "matchTimeInUnknown": 60.000,
```

```
                "matchDurationInUnknown": 33.000,
                "matchTimeInReference": 0.500,
                "metadata": {
                      "ContentCategory": "Music",
                      "Title": "Feels Like Summer",
                      "Artist": "Weezer",
                      "ISRC": "USAT21700604",
                      "Vendor": "WMG",
                      "AlbumReleaseDate": "01-Jan-1900",
                      "AlbumUPC": "Unspecified",
                      "SongLengthInSeconds": "195"
                }
          }
    ],
    "territoryRules": {
          "AD": "block",
          "AE": "block",
          "AF": "block",
          ...
          [entries omitted here to save space in this example]
          ...
          "ZA": "block",
          "ZM": "block",
          "ZW": "block"
    }
}
```


analyze

Usage

The *analyze* command-line program is run like this:

analyze -c *configFile* -i *mediaFile* -e *unknownAssetID* -o *outputFile*

- *configFile* is the configuration file supplied to you by Audible Magic.
- *mediaFile* is the media file you wish to identify.
- *unknownAssetID* is a customer-determined string that identifies the media file in the customer's systems.
- *outputFile* is a binary file to be ingested later by *identify*.

Example

Say you want to identify the same media file mentioned above, containing *Feels Like Summer* by Weezer. You could first analyze the media file with the following command:

```
analyze -c my.config -i 82345.mp4 -e 82345 -o 82345.analysis
```

Then, at a later time, or from a different computer, you could send an ID request to our servers and get back the result:

```
identify -c my.config -i 82345.analysis -o 82345.json
```

The configuration file should be the same one that was fed to *analyze.* The output JSON file would be expected to be identical to the one generated if you had simply fed *identify* the media file instead of the .analysis file, as in the earlier example of how to use *identify.*

Program output to stderr and stdout

In addition to the JSON output, the command-line programs described above will output error information to stderr as a convenience to the human operator of these programs. When the -v (verbose) flag is specified, additional information is sent to stdout. We expect that most users of the software will be running many identifications and will automate the process with a script that will inspect the generated JSON file and take further action.

Both programs include additional options. Type the program's name followed by `-?` for details.

JSON output specification

status code

The JSON output will always include a **statusCode** (an integer value) and a **statusDescription.**

statusCode	Meaning
2005	No matches were found.
2006	At least one match was found.
2045	Audible Magic took too long to respond to the request.
5429	HTTP status code 429 – too many requests. See "Design Considerations – Retries and Timeouts" below for more information.
All others	Contact Audible Magic Client Services if the error is not self-evident.

statusDescription

A brief explanation of the status code. Besides a textual explanation, it includes the status code and another number in parentheses which might differ. If you need help understanding an error condition, provide the full description, including both numbers, to Audible Magic's Client Services.

unknownAssetID

The JSON output echoes the string **unknownAssetID** that was passed with the -e flag to the *identify* program, except in some error conditions such as when the command line was incorrect.

matches

If Audible Magic determines that portions of the media file do match references in the Audible Magic database, a **matches** section will be included. This is a JSON array of match structures, each of which usually contains:

- **ruleCode** – what action to take (e.g., "block", "allow"). If this field is omitted, there will instead be a **territoryRules** section outside the **matches** section (see below).
- **matchType** – what kind of reference was matched (e.g., "audio", "video")
- **amItemID** - Audible Magic's internal asset ID for the matched reference
- **matchTimeInUnknown** - the location in seconds of the start of the match in your media file
- **matchDurationInUnknown** - the duration of the match in seconds
- **matchTimeInReference** - the location of the start of the match in Audible Magic's original reference media file
- **metadata** - a set of key-value pairs associated with the match. It will always include one key named **ContentCategory**, and the value of this key determines what other metadata keys are included.

If **ContentCategory** is "Music", the other metadata keys are:

- **Title** - the title of the reference given by the content supplier
- **Artist** - the artist of the reference given by the content supplier
- **ISRC** - the Industry Standard Recording Code given by the content supplier (this field is present for some Toolkit customers)
- **Label** - the label associated with the copyright holder
- **Vendor** - the supplier of the reference material
- **AlbumReleaseDate** - when the album containing the reference material was released
- **AlbumUPC** - the universal product code of the album that the reference material was released on
- **SongLengthInSeconds** - the length of the reference recording as submitted by the supplier

If **ContentCategory** is "Video" or "Soundtrack", the other metadata keys are:

- **Title** - the title of the reference given by the supplier
- **Studio** - the film studio associated with this reference
- **Vendor** - the supplier of the reference material

territoryRules

This section is included for all customers who do not receive a **ruleCode** in the **matches** section (see above). It is present only when a match was made (i.e., when **statusCode** is 2006). **territoryRules** is a list of generally over 200 lines, each specifying an ISO territory code (such as "US" for the United States or "JP" for Japan) and the action that should be taken when the user is located in that territory. In the example above, the action ("rule") was always "block," meaning that the content owner has specified that the content should be not be made available in any territory of the world. Other possible rules include "allow", "monetize", and "track".

Sample JSON output files

1. Erroneous command line

This sample shows a JSON output file when the command was missing needed information. (Details about what was wrong are generally available on stderr.)

```
{
  "statusCode": 3,
  "statusDescription": Error 3 (3) Incorrect command line
}
```

2. No match

This sample shows output JSON when the ID request was successfully processed, but no matching content was discovered, for a media file with unknownAssetID 43259XX:

```
{
    "statusCode": 2005,
    "statusDescription": "Status 2005 (2005) No match found",
    "unknownAssetID": "43259XX"
}
```

3. Match, with limited metadata

This sample shows the output JSON when matching content was discovered. This JSON contains the limited metadata for a customer of Audible Magic's RightsRx™ service:

```
{
    "statusCode": 2006,
    "statusDescription": "Status 2006 (2006) Match found",
    "unknownAssetID": "297J93",
    "matches": [
        {
            "ruleCode": "block",
            "matchType": "audio",
            "amItemID": "95f8258948da4026ac19db418a46c8be_LBTa00",
            "matchTimeInUnknown": 166.950,
            "matchDurationInUnknown": 32.000,
            "matchTimeInReference": 154.529,
            "metadata": {
                "ContentCategory": "Music",
                "Vendor": " Orchard",
            }
        }
    ]
}
```


4. Match, without territory rules

This sample shows output JSON for a compliance customer who does not receive per-territory rules. As in the previous sample, the **matches** structure contains a **ruleCode** field, but in this case more metadata is included:


```
{
    "statusCode": 2006,
    "statusDescription": "Status 2006 (2006) Match found",
    "unknownAssetID": "TU8-94-7.mp3",
    "matches": [
        {
            "ruleCode": "block",
            "matchType": "audio",
            "amItemID": "4514a672110f4a4eb96e507b33c592fd_RGUa00
            "matchTimeInUnknown": 0.500,
            "matchDurationInUnknown": 33.000,
            "matchTimeInReference": 0.500,
            "metadata": {
                "ContentCategory": "Music",
                "Title": "Feels Like Summer",
                "Artist": "Weezer",
                "ISRC": "USAT21700604",
                "Vendor": "WMG",
                "AlbumUPC": "Unspecified",
                "SongLengthInSeconds": "195"
            }
        }
    ]
}
```

5. Multiple matches, along with territory rules

In this case, the lookup resulted in multiple matches. This sample also shows a **territoryRules** section, rather than a single **ruleCode** in the **matches** section. (The presence of territory rules is independent of whether there are multiple matches, however; it instead depends on what service Audible Magic has set up the customer for.)


```
{
    "statusCode": 2006,
    "statusDescription": "Status 2006 (2006) Match found",
    "unknownAssetID": "2JJJ89",
    "matches": [
        {
            "matchType": "audio",
            "amItemID": "bd846e703f254e3187d87941b6381f38_CGPa00",
            "matchTimeInUnknown": 52.500,
            "matchDurationInUnknown": 305.000,
            "matchTimeInReference": 6.897,
            "metadata": {
                "ContentCategory": "Music",
                "Title": "Thriller",
                "Artist": "Michael Jackson",
                "ISRC": "USSM10501511",
                "Label": "Epic",
                "Vendor": "Sony",
                "AlbumUPC": "A10301A0000622739MCG",
                "SongLengthInSeconds": "313"
            }
        },
        {
            "matchType": "audio",
            "amItemID": "95f8258948da4026ac19db418a46c8be_LBTa00",
            "matchTimeInUnknown": 13.500,
            "matchDurationInUnknown": 290.000,
            "matchTimeInReference": 0.500,
            "metadata": {
                "ContentCategory": "Music",
                "Title": "Thriller",
                "Artist": "The Freaks",
                "ISRC": "NLG620538813",
                "Label": "EMD",
                "Vendor": "Orchard",
                "AlbumReleaseDate": "2014\/02\/10",
                "AlbumUPC": "8718781128203",
                "SongLengthInSeconds": "297"
            }
        }
    ],
    "territoryRules": {
         "AD": "block",
         "AE": "block",
        "AF": "block",
         ...
         [entries omitted here to save space in this example]
         ...
         "ZA": "block",
         "ZM": "block",
```

```
            "ZW": "block"
        }
    }
```

Service Notes

Some important things to note about the service and testing:

1) This service requires that at least 22 seconds of audio be sent in for a standard-resolution lookup or at least 6 seconds of audio for a high-resolution type lookup. (See the *Design Considerations* section below.)

2) If you are working in a test environment, rather than a production setting, please only send 1 concurrent transaction at a time, because Audible Magic's service for testers is shared with other customers. If you exceed this limit, the Toolkit may return the error "HTTP status code 429 – too many requests." Once you are in a production environment, the maximum number of concurrent transactions is set contractually.

Design Considerations

Supported Formats and Codecs (non-mobile platforms)

The desktop/server version of the library can support a wide variety of audio and video file formats and codecs. It uses open-source software from the FFmpeg project and, in general, it will handle any kind of file that the FFmpeg executable supports. For a complete list please see https://www.ffmpeg.org/general.html#Supported-File-Formats_002c-Codecs-or-Features. If you have questions about a specific format or codec, or have difficulty using it with the Toolkit, please contact Audible Magic Support.

NOTE: *Files protected by Digital Rights Management (DRM) cannot be fingerprinted.*

For best identification results, the unknown audio should be at least moderately high-quality, such as 128-kbps MP3 files, WAV files containing 22.5-kHz 16-bit samples, or similar.

Retries and Timeouts

Customers of the Audible Magic Toolkit should equip their applications to handle service availability issues. In the event of a connectivity or other error, the application must evaluate what should be done with unprocessed requests. For example, sites with large amounts of queries may elect to queue up the transactions to resubmit when service is reestablished. There are some error situations (e.g., network problems, server over capacity) where the server will return an error and the transactions should be resubmitted.

If ID requests are called too frequently by simultaneous processes, you might find that the statusCode in the Toolkit's JSON output is 5429 ("HTTP status code 429"). HTTP 429 means "too many requests," indicating you have exceeded the maximum number of transactions per minute allowed in your contract with Audible Magic. When it receives a HTTP 429, the identify program will attempt to resend the request repeatedly, with an increasing amount of delay between attempts. If all these attempts receive the HTTP 429, identify will then exit with 5429 in the JSON. If this happens, reduce the number of simultaneous processes that are sending requests to Audible Magic's servers and/or how frequently each runs the identify program.

Other methods to avoid a 5429 error are to:

- Include a process in your code that regulates the rate of your requests so that they are distributed evenly over time.
- Implement a queuing mechanism where transactions go into a queue to be processed and limit your handlers to be fewer than maximum number of allowed transactions per period of time.

If you are regularly approaching or bumping into the rate limit and feel that the number is not meeting your needs, please contact helpdesk@audiblemagic.com. A team member will help assess your requirements and may suggest alternatives.

Reporting a Problem

It is our goal to provide Audible Magic Toolkit users with helpful and timely customer support. We are available to answer any questions you may have and to provide technical assistance should you encounter any problems with the Audible Magic Toolkit.

Should you require assistance, submitting a helpdesk ticket is the best way to contact our support team. This can be done by either sending an email to helpdesk@audiblemagic.com or via the support portal at http://support.audiblemagic.com.

Audible Magic Client Services handles general requests during regular business hours. Monday-Friday 9:00AM– 5:00PM Pacific Time. Response time is within 8 business hours.

When you report a problem to Audible Magic there are several questions we will ask. Answering these questions when submitting a ticket will quicken the response time to your problem. The more of the questions below that you can answer, the more Audible Magic Client Services will understand your problem and be able to help you solve it.

Problem Description

Please provide a concise description of the problem encountered. Be sure to say what you expected to happen, and what actually happened. Characterize what the application is doing when the failure (or

unexpected result) occurs.

Please include:

- The complete Toolkit command line you are trying to execute.
- The Audible Magic configuration file.
- The identify program's complete JSON output, or at least the statusDescription field.
- Any error message displayed following your command line or written to stderr.

Environment

- What version of the Toolkit are you using (e.g., 38.11)? (If unsure, run "identify -?" and look for "Audible Magic library version.") Also include the Toolkit release package's file name if possible.
- What platform are you running on when the problem is encountered (e.g, Ubuntu, Red Hat Enterprise Linux, Windows)?
- What specific version of the platform are you using (e.g., Ubuntu 16.04, Windows 10 Pro, macOS 10.12.5)?

Problem Duplication

- Is the same error or problem encountered on every run of the same command, or is it inconsistent?
- Is the problem encountered when using a specific input media file?
- Does the problem occur when using some input media files(s) but not others? If input media files are involved, Audible Magic Client Services may request temporary access to them for debugging purposes.
- If the error is reported by the identify program, do you see the same error when feeding the same media file and configuration file to the analyze program? If not, there might be a communication problem.
- If you have previously used an older Audible Magic Toolkit or SDK, do you see the problem when using it to run the equivalent command?
- Do you see the same behavior if you install the Toolkit on a different machine?
- Do you see the same behavior if you install the Toolkit on a different operating system?