ID3 Tag Validator Application

User Guide

Release 1.8 Revision F

Detailed Revision History

Revision	Date	Changes Made	Responsible Engineer
A	2015-01-26	Initial Draft	Lois Price
B	2015-05-15	Revised for Release 1.5	Lois Price
Pre-release	2016-02-03	Revised for beta release of 1.7	Lois Price Lore Eargle (editor)
C	2016-02-08	Production release of 1.7	Lois Price Lore Eargle (editor)
D	2017-04-05	Internal Release of 1.8: added MPEG-DASH	Lois Price Lore Eargle (editor)
E	2017-06-29	Updated Figure 3 – Summary Reports Contents and Figure 5 – SID List Contents	Lois Price Lore Eargle (editor) Harish GP (reviewer)
F	2017-09-27	Clarified examples of –d, -df, and –dr input segment designators	Lois Price Lore Eargle (editor) Harish GP (reviewer) Julia Wen (reviewer)

Contents

List of Figures

List of Tables

Overview

Purpose

As part of its strategy to measure audience viewing of content on consumer devices, Nielsen supports or endorses a variety of products that generate Nielsen ID3 tags and insert them into segmented MPEG-2 transport streams or into MPEG-DASH containers. The Nielsen Validator application (referred to as “Validator”) is a diagnostic tool that analyzes Nielsen ID3 tags and helps in troubleshooting problems with applications that generate such tags.

These are six principal ways to use Validator:

- Process a transport stream (TS) with embedded Nielsen ID3 tags (option –t)
- Process an HLS stream with the m3u8 index and transport segments zipped into a file (option –m)
- Process a file of encrypted ID3 tags such as the output of the PCM-to-ID3 SDK sample application or the log file of a mobile device that processes Nielsen ID3 tags (option –r)
- Process specific MPEG-DASH segments in a folder, guided by the user’s command-line instructions, to guarantee that the segments are processed in the proper order (-dr option – this is the preferred method of processing MPEG-DASH content)
- Process a set of MPEG-DASH segments that have been merged into a single file (option –d)
- Process all MPEG-DASH segments in a specific folder that end with a user-defined extension (option–df)

This manual explains how to use Validator as a diagnostic tool. It focuses on the testing and debugging of applications that use the PCM-to-ID3 SDK to generate Nielsen ID3 tags. It describes issues that commonly arise during the integration process and how to resolve them.

In addition, developers who create applications based on the Nielsen ID3 Tag Validator SDK can use Validator prior to sending their test files for transcoder certification.

Customer Support

Your Nielsen support team welcomes any questions about Validator and its output.

Intended Audience

This manual is written for the following users:

- A software developer who has integrated PCM-to-ID3 SDK into an application that supports Nielsen ID3 Tag generation and who may or may not be submitting their application or device for Nielsen certification
- A quality-assurance professional who is testing an application into which the PCM-to-ID3 SDK has already been integrated

Terminology

To properly interpret the output of Validator, you should be familiar with the basic components of the Nielsen ID3 tag. For an in-depth description of a typical ID3 tag, see "Detailed Tag Listing – DATA Tag" on page 43.

This manual uses a number of other terms and acronyms that are specific to the process of generating Nielsen ID3 tags. For definitions, see "Glossary" on page 62.

Related Documents

- Apple. iOS Developer Library, "Timed Metadata for HTTP Live Streaming." https://developer.apple.com/library/ios/documentation/AudioVideo/Conceptual/HTTP_Live_Streaming_Metadata_Spec/2/2.html
- *ID3 Tag Validator SDK Developer Guide:* provides instructions on how to create applications that analyze and evaluate Nielsen ID3 tag streams
- ISO/IEC 23009-1:2014(E): Information technology—Dynamic adaptive streaming over HTTP (DASH), Part 1: Media presentation description and segment formats specification. See Section 5.10.3.2 for a discussion of MPEG-DASH Media Presentation Description (MPD) signaling of in-band events.
- *PCM-to-ID3 Tag SDK Developer's Guide*: describes how to use the Nielsen PCM-to-ID3 SDK to generate Nielsen ID3 tags. It provides an overview of the format of Nielsen ID3 tags and of their role in measuring audience viewing on a variety of consumer devices.
- *Transcoder SDK Certification Process – Vendors*: describes the test files for implementation of the Nielsen ID3 SDK and instructions on how to run the tests

Using Validator in the Transcoder Certification Process

Use of Validator for certification of a transcoder is based on two assumptions:

- Your software application follows the process outlined in the Nielsen *PCM-to-ID3 Tag SDK Developer's Guide*.
- The output of your application is one of the following:
  - A zipped package holding an HLS stream that includes Nielsen ID3 tags
  - A zipped package holding an MPEG-DASH stream where Nielsen ID3 tags are stored in emsg boxes. The emsg boxes should be attached to audio segments.

Validator is the primary tool used in evaluating the files that you deliver to Nielsen for analysis. Before delivering the test files to Nielsen, you may use Validator to process the files yourself.

Note the following cautions and contact your Nielsen support team with any questions:

- For certification tests, a failure of the CID SID test is not unusual. If most of the CID SID errors listed in the log file or noted in the Comma-Delimited Tag Listing are associated with the watermark SID transitions that periodically occur, ignore the errors. See “Test 6 Result: CID SID Test” on page 26.
- You may see occasional watermark decoding errors that result in invalid EDU SIDs or timestamps. Occasional watermark errors are to be expected. An *excessive* number of watermark SID or timestamp warnings, however, may result from faulty audio decoding.

Pre-Validation Checklist

A common source of tag-generator errors is the misinterpretation or omission of one or more of the processes listed in Table 1. To avoid these mistakes and save yourself some time, review these steps.

Table 1 – Pre-Validation Checklist

Step	Process	Requirements	Note
1	Your application receives a source MPEG-2 transport stream that contains Nielsen watermarks	Ensure the source content is watermarked.	
2	Your application extracts the audio stream from the source transport stream and converts the audio to PCM format.		If the source program contains more than one audio elementary stream, your application selects the first stream in the Program Map Table (PMT).
3	Your application delivers buffers of PCM audio data to the PCM-to-ID3 SDK and stamps each buffer with the "current time."	Ensure the "current time:" advances by 1 second for every second of audio processed.	**Potential and most frequent error:** The application feeds system clock time to the ID3 SDK when it processes the incoming audio content faster than the rate at which the audio is normally delivered during live streaming.
4	The PCM-to-ID3 SDK decodes the Nielsen watermarks in the incoming audio stream, generates ID3 tags, and delivers ID3 tags and log messages to your application	Ensure your application stores the log messages in a retrievable log file.	
5	**ID3 Tags in HLS Steams Only** Your application inserts each Nielsen ID3 tag into two transport packets in a single metadata Packetized Elementary Stream (PES) packet.	Ensure the PMT elementary loop defining the metadata PES includes the required ID3 tag descriptors, as defined in "Timed Metadata in HTTP Live Streaming." Within the metadata PES, the two transport packets carrying the ID3 tag must be contiguous.	See "Related Documents" on page 7.

Step	Process	Requirements	Note
	ID3 Tags in MPEG-DASH Steams Only At the start of processing, your application generates an MPD that includes the line <InbandEventStream schemeIdUri="www.nielsen.com:id3:v1" value="1"/>		
6	**ID3 Tags in HLS Steams Only** Your application stamps the metadata PES packet with the Presentation Timestamp (PTS) of the most recently processed audio PES packet.	Apply the proper PTS to the metadata PES packet to synchronize the ID3 tags with the corresponding audio.	
	ID3 Tags in MPEG-DASH Steams Only Your application inserts the entire Nielsen ID3 tag in an emsg box.	See section 5.10.3.3 of ISO / IEC 23009-1-2014(E) for a detailed explanation of emsg box format and signaling. The DASHEventMessageBox scheme_id_uri should be set to the null-terminated string “www.nielsen.com:id3:v1”. The ‘value’ field must be set to the null-terminated ASCII string “1”. The message_data field must hold the entire Nielsen ID3 tag.	The entire 271-byte ID3 tag delivered by PCM-to-ID3 SDK (step 4) must be included, without alteration, in the emsg box. Eliminating or modifying any portion of the ID3 tag may result in crediting errors.
7	**ID3 Tags in HLS Steams Only** Your application multiplexes the metadata PES packet into the outgoing MPEG stream.		
8	**ID3 Tags in HLS Steams Only** Your application delivers the transport stream to an HLS segmenter.		The segmenter splits the transport stream into multiple transport-stream (TS) segments and generates the required m3u8 index.
9	The segmented content is delivered downstream to mobile devices and/or browsers.		

Figure 1 is a visual representation of the tagging process for HLS and MPEG-DASH Streams. The green boxes represent the functionality your application provides. The blue box represents functionality that the Nielsen PCM-to-ID3 SDK provides.

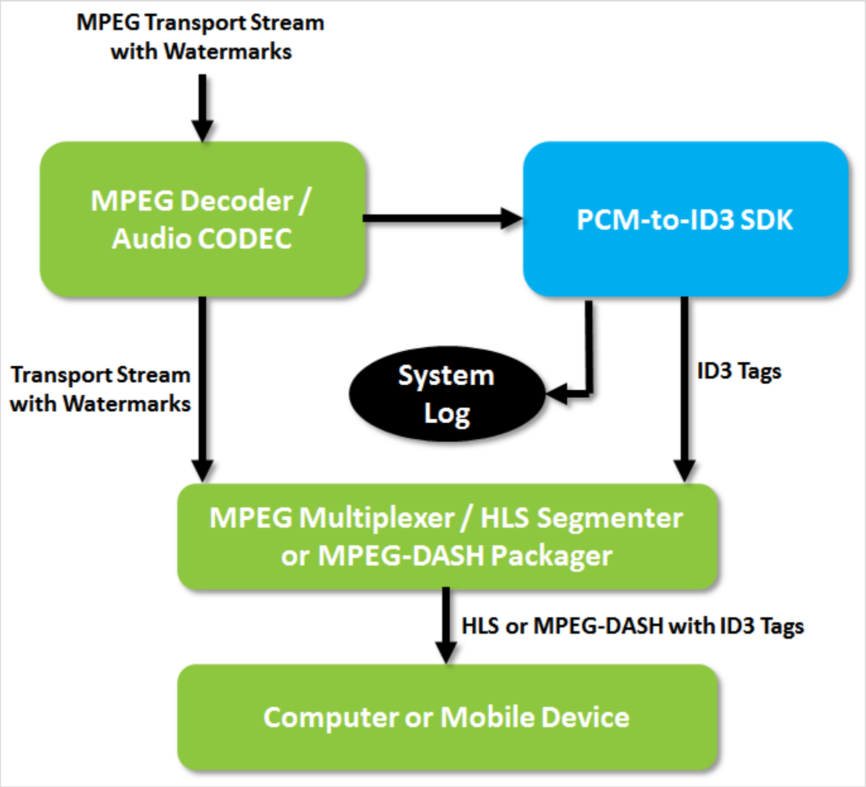

MPEG Transport Stream with Watermarks
MPEG Decoder / Audio CODEC
PCM-to-ID3 SDK
Transport Stream with Watermarks
System Log
ID3 Tags
MPEG Multiplexer / HLS Segmenter or MPEG-DASH Packager
HLS or MPEG-DASH with ID3 Tags
Computer or Mobile Device

Figure 1 – Tagging Process

Running the Validator Application

Basic Instructions

Validator is typically delivered as a Microsoft® Windows® console application. Follow these steps to run it.

1. Open a command prompt.
2. Change directories to the location of the Nielsen ID3 Tag Validator executable.
3. Determine which of the following file types you want to validate and type the corresponding command that follows:
    - Use **–r** for a log file in the format generated by the Nielsen sample application installed on your computer or mobile device or any other file that holds encrypted Nielsen ID3 tag payloads (249 bytes, starting with “www.nielsen.com”):

      `NielsenId3TagValidator -r` *<full path name of file holding the ID3 tags>* **`-p 6`**

      Note Validator locates and analyzes all ID3 Tags in the log file and ignores all text entries between tags.

    - Use **–t** for a transport stream with Nielsen ID3 tags “muxed” into it:

      `NielsenId3TagValidator -t` *<full path name of the transport stream>* **`-p 6`**

    - Use **–m** for a set of transport-stream segments, accompanied by an HLS m3u8 index file:

      `NielsenId3TagValidator -m` *<full path name of the m3u8 file>* **`-p 6`**

    - Use **–dr** for a folder holding MPEG-DASH segments, where some of the segments hold emsg boxes containing Nielsen ID3 tags. Use this call to **guarantee** that segments are processed in the correct order:

      `NielsenId3TagValidator -dr` *<folder name>* **`-b`** *<base segment name>* **`-e`** *<extension>* **`-first`** *<index of first segment>* **`-last`** *<index of last segment>* **`-i`** *<index interval between segments***`> -p 6`**

    Example: Folder *C:\test* has ten MPEG-DASH segments with the following file names:

    test12**10**9tmp.m4s

    test12209tmp.m4s

    test12309tmp.m4s

    test12409tmp.m4s

    test12509tmp.m4s

    test12609tmp.m4s

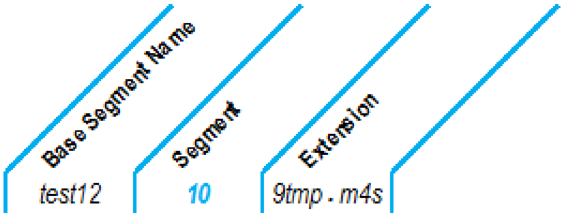
test12709tmp.m4s

test12809tmp.m4s

test12909tmp.m4s

test121009tmp.m4s

The file names are in the format *<base segment name><segment><extension>* as follows:



First segment	*test12109tmp.m4s*
Last segment	*test121009tmp.m4s*
Segment increase interval	*10*

To process these MPEG-DASH segments, use the following command:

```
NielsenId3TagValidator -dr C:\test -b test12 -e 9tmp.m4s -first 10 -
last 100 –i 10 -p 6
```

- You can also use either the **–d** or **–df** commands. The **–d** command requires that the segments be in order. For the **–df** command, there is no guarantee that the segments will be presented in order. The **–dr** command described above, therefore, is the best option.

  - For a single file holding a set of MPEG-DASH segments, where some of the segments hold emsg boxes containing Nielsen ID3 tags, you can use the **-d** command as follows:

    ```
    NielsenId3TagValidator -d <full path name of file> -p 6
    ```

  - For a folder holding MPEG-DASH segments, where some of the segments hold emsg boxes containing Nielsen ID3 tags, you can use the **-df** command. The tag listing displays the name of each segment followed by the string, *** *EMSG: www.nielsen.com:id3:v1*, before it lists the tags found in that segment.

    Note Within the folder, the segments must be arranged in such a way that the copy /b command would gather them in the order in which they would normally be played back by a DASH player.

    ```
    NielsenId3TagValidator -df <folder name> -e <file extension of segments> -p
    6
    ```

    Example: To process all MPEG-DASH segments with extension *.m4a* located in folder *C:\test*, use this call:

    ```
    NielsenId3TagValidator -df C:\test -e .m4a -p 6
    ```

Optional Parameters

The only required command-line argument is the input-type designator, which has a value of -r, -t, -m, -d, -df, or -dr. In addition, there are three optional command-line parameters: -o, -p, and -n.

Output Folder Specifier (-o)

The **-o** option allows you to specify the folder where your output files should be stored. **Do NOT end the path name with a trailing slash (\).** If you do not use the -o option, Validator stores the output files in the same folder as the input file.

Report Type Specifier (-p)

The **-p** option allows you to generate one or more of the following reports:

- **Summary Report:** always generated, regardless of -p setting, this reports the following:
  - INFO tag fields
  - Breakout type
  - List of PC (Program Content) and FD (Final Distributor) SIDs
  - Results of the 30 tests described in “Test Results” on page 21
- **Log File:** always generated, regardless of -p setting, the file contains the following:
  - Reports on errors that Validator detects
  - Lists the sequence number and a helpful error message for each
  - Where applicable, lists the structure and contents of the transport stream
- **Detailed Tag Listing:** reports all fields of the decrypted ID3 tag with multiple lines per tag
- **Comma-Delimited Tag Listing:** summarizes content of a single ID3 tag in a single line with comma-delimited fields. The best way to use this data is to import this file into a spreadsheet.

For analysis of content *stored in files*, use any of the -p settings shown in Table 2.

Table 2 – -p Option Settings

-p Setting	Summary Report	Log File	Detailed Tag Listing	Comma-Delimited Tag Listing	Periodic Report*
0 (default)	x	x			
1	x	x			x
2	x	x	x		
3	x	x	x		x

-p Setting	Summary Report	Log File	Detailed Tag Listing	Comma-Delimited Tag Listing	Periodic Report*
4	x	x		x	
5	x	x		x	x
6	x	x	x	x	
7	x	x	x	x	x

Program Number Specifier (-n)

It would be unusual for a transcoder to deliver its final output as a multi-program transport stream. However, if you do happen to ask Validator to process a multi-program transport stream (MPTS), you may want to designate a specific program to be processed. Each program has its own set of ID3 tags. Use the **-n** option to specify the program number of the program with the ID3 tags that Validator should analyze. For example, -n 2 instructs Validator to report only on ID3 tags associated with program 2 in a multi-program transport stream.

Note that Validator lists identifying information for all programs in the PAT and PMT, even if the program number does not match the program number that you provide. Validator, however, parses only the metadata stream associated with the program that you specify.

Using Validator Output

There are three possible outcomes from running Validator:

- All tests pass, and the error count for each test is acceptably low.
- Some or all of the test reports include an excessive error count in the first field. To check these, do one or both of the following tasks:
    - Examine the log file and Detailed Tag Listing. See “Detailed Tag Listing – DATA Tag” on page 43.
    - Import the Comma-Delimited Tag Listing into a spreadsheet and note the error messages in the last column. For specifics, see and “Comma-Delimited Tag Listing” on page 50.

Validator Output: Summary Report

Overview

Validator always generates the two output files listed in Table 3.

Table 3 – Default Output Files

File	Ends With	Description
Summary Report	_report.txt	Summarizes the results of the ID3 tag analysis
Log file	_log.txt	Lists messages about errors or anomalies found in the ID3 tag stream

Do the following:

1. To determine whether the source ID3 tag stream contains errors, review the Summary Report.

   The code labeled as Figure 2 below shows an example of a Summary Report generated when the application was run with the -m option. The Summary Report has four major components, which are described in detail in the sections that follow this one:

   - File header
   - SID list
   - INFO tag list
   - 30 pass/fail test reports

2. To see more details on any errors, review the log file.

```
VALIDATOR SUMMARY REPORT, Version 1.8
File evaluated: stress_test.m3u8
Duration: 915 seconds
Breakout Type: 0

FD SID  9120: NAES 2 count = 129, NAES 6 count = 114
PC SID  9020: NAES 2 count = 111, NAES 6 count = 103
PC SID  9051: NAES 2 count = 108, NAES 6 count = 106
FD SID  9151: NAES 2 count = 128, NAES 6 count = 127

INFO TAG:
Vendor ID: 999
Device Type: Transcoder or related device
Device ID: C81F66B8057C
SystemVersion: 1234
SDK Version: 2.3
```

```
Audio CODEC: AAC 2.0 64kbps
Distributor ID: www.example.com

TEST RESULTS:
DATA Tag : DATA Tag Count:            Pass <105/81/0>
DATA Tag : Tag Format Test:           Pass <0/0/11>
DATA Tag : Sequence Number Test:      Pass <0/0/2>
DATA Tag : Time Gap Test:             Pass <0/0/2>
DATA Tag : Tag Duration Test:         Pass <0/0/3>
DATA Tag : CID SID Test:              Pass <3/0/11>
DATA Tag : CID Timestamp Test:        Pass <0/0/11>
DATA Tag : CID TS Offset Test:        Pass <3/0/11>
DATA Tag : Breakout Type Test:        Pass <0/0/1>
DATA Tag : Watermark Timecode Test:   Pass <30/0/57>
DATA Tag : Watermark Level Test:      Pass <19/0/57>
INFO Tag : INFO Tag Count:            Pass <4/2/0>
INFO Tag : INFO Tag Interval Test:    Pass <0/0/1>
INFO Tag : INFO Tag CID Test:         Pass <0/0/1>
INFO Tag : INFO Tag Offset Test:      Pass <0/0/1>
INFO Tag : Vendor ID Test:            Pass <0/0/1>
INFO Tag : Device Type Test:          Pass <0/0/1>
INFO Tag : Device ID Test:            Pass <0/0/1>
INFO Tag : System Version Test:       Pass <0/0/1>
INFO Tag : SDK Version Test:          Pass <0/0/1>
INFO Tag : Distributor ID Test:       Pass <0/0/1>
MPEG - 2 : Tag Placement Test:        No Status <0/0/0>
MPEG - 2 : Tag Time Test:             Pass <0/0/6>
MPEG - 2 : PES Packet Header Test:    Pass <109/81/0>
MPEG - 2 : Nielsen Owner ID Test:     Pass <109/1/0>
MPEG - 2 : Missing PES PTS Test:      Pass <0/0/6>
MPEG - 2 : PES Header Test:           Pass <0/0/6>
MPEG - 2 : PES Incomplete Tag Test:   Pass <0/0/3>
MPEG - 2 : PMT Descriptor Test:       Pass <98/1/0>
MPEG - 2 : PMT Elem Stream Test:      Pass <98/1/0>
```

Figure 2 – Summary Report Contents

File Header

Description and Example

The Summary Report header identifies the name of the input file, specifies the length of the analyzed content, and lists the breakout type. If you used the -m option when you ran Validator, the Summary Report header looks similar to that shown in Figure 4.

```
VALIDATOR SUMMARY REPORT, Version 1.8
File evaluated: index.m3u8
Duration: 918 seconds
Breakout Type: 0
```

Figure 3 – Summary Report File Header

Diagnostic Tips

- Verify the Validator version number is correct.
- Check the “File evaluated” line to ensure that the report refers to the expected file or set of segments. See “Basic Instructions” on page 12.

- For the **-m** option, the “File evaluated” line should specify an HLS m3u8 index file as shown in Figure 2.
- For the **-r** option, the “File evaluated” line should be an mp4 file that has embedded tags or an encrypted tag file such as a file extracted from a mobile-device log running the Nielsen sample application.
- For the **-t** option, the “File evaluated” line should refer to an MPEG-2 transport stream.
- For the **-d** option, the “File evaluated” line should refer to a set of MPEG-DASH segments merged, in the correct order, into a single file.
- For the **-df** option, the “File evaluated” line should hold the folder name, followed by “MpegDash.txt.” Note that there is no file named “MpegDash.txt.” The name is simply a placeholder. The folder name, however, should refer to the folder that you named after the -df option.
- For the **-dr** option, the “File evaluated” line should hold the base name (including the folder name) of the segments that were processed. When stripped of the path name, this base segment name should match the string that you entered on the command-line after the -b option.

- Check the “Duration” line to make certain that the PCM-to-ID3 SDK generated tags for all of the watermarked content that it processed.

  Note The reported duration (in seconds) is calculated by subtracting the tag time of the first decoded ID3 tag from the tag time of the last decoded ID3 tag.

  If the beginning or the end of the processed file has no Nielsen watermarks, the reported duration is less than the number of seconds of processed audio/video content. Otherwise, the duration listed in the Summary Report header should be within 20 seconds of the duration of the processed content.

- Make certain that the reported breakout type matches the setting that you used when you configured your application. Your application should always set the breakout code to 0.

SID List

Description

The SID List identifies the PC (Program Content) and FD (Final Distributor) SIDs (source identifiers) that Validator found in the ID3 tags in the analyzed content. Validator lists only the SIDs that it associates with the PC or FD code. Due to bit errors during the decoding process, invalid SIDs may appear in the SID List, but their counts are significantly less than those of valid SIDs. SIDs with a “UK” (unknown) watermark level or distribution type may not appear in this list at all, even if the same SID is later identified as a PC or FD SID.

In the SID List in Figure 4, the valid FD SID is 9151, and the valid PC SID is 9051. In the entire file, the level was incorrectly interpreted as PC instead of FD for five unique NAES 2 audio codes.

```
FD SID  9151: NAES 2 count = 160, NAES 6 count = 166
PC SID  9151: NAES 2 count = 5 , NAES 6 count = 0
PC SID  9051: NAES 2 count = 110 , NAES 6 count = 120
```

Figure 4 – SID List Contents

Note that some streams may have up to three different *FD SID*s in the same file segment. However, there should be only one *PC SID* in a single portion of a specific file. “Single portion” refers to a commercial, a program segment, etc.

Table 4 shows the interval in seconds at which we would ideally expect to see a specific SID within any segment of the stream with the listed characteristics. The first three rows in the table show the most common configurations.

Table 4 – Interval in Seconds

	NAES 2		NAES 6			
Segment Type	PC	FD	PC	FD1	FD2	FD3
PC-Only	4	N/A	4.8	N/A	N/A	N/A
PC + FD	4	4	4.8	4.8	N/A	N/A
FD Only	N/A	2	N/A	4.8	N/A	N/A
FD+FD	N/A	2	N/A	4.8	4.8	N/A
FD+FD+FD	N/A	2	N/A	4.8	4.8	4.8
PC+FD+FD	4	4	4.8	4.8	4.8	N/A

Examples

For a 15-minute segment of audio that has only a PC SID of 100, expect to see a SID List similar to this:

```
PC SID  100: NAES 2 count = 225, NAES 6 count = 187
```

For a 15-minute segment of audio that has a PC SID of 100 and an FD SID of 200, expect to see a SID List similar to this:

```
PC SID  100: NAES 2 count = 225, NAES 6 count = 187
FD SID  200: NAES 2 count = 225, NAES 6 count = 187
```

For a 15-minute segment of audio that has only an FD SID of 200, expect to see a SID List similar to this:

```
FD SID  200: NAES 2 count = 450, NAES 6 count = 187
```

Diagnostic Tips

In evaluating the SID List, remember that the process of encoding and decoding NAES watermarks does not always produce ideal results. The following factors may significantly reduce the number of NAES watermarks detected:

- The audio is “quiet,” such as during a talk show.
- Compression or poor resampling techniques produce audio distortions that negatively affect NAES watermark detection.
- NAES encoding allows one FD SID to overwrite an FD SID previously inserted into the stream. For example, FD SID 100 may have been inserted at the broadcast facility where the content originated; afterwards, an encoder at a local broadcast affiliate may overwrite SID 100 with FD SID 200. If the SIDs are not precisely aligned during the overwriting process, the decoder may incorrectly decode portions of the watermark (SID, timestamp).
- The audio is clipped at some frequencies.

Whenever the NAES decoder within that PCM-to-ID3 SDK is unsuccessful in decoding PC or FD watermarks, the ID3 tag inserter may be thwarted in its ability to generate ID3 tags. If no watermarks *appear* for the entire 10-second span of an ID3 tag, no ID3 tag is *generated* for that 10-second period. Therefore, a reduced SID count in the SID List of the Summary Report may well be reflected in a reduced tally of DATA tags. See “Test 1 Result: DATA Tag Count” on page 21.

INFO Tag Listing

Description and Example

The INFO Tag listing is the third section in the Summary Report. It identifies all of the INFO-tag fields. Interspersed among the DATA tags at 5-minute intervals, INFO tags identify your device or application as the source of the DATA tags. Validator simply lists the value of each of the fields. It declares an error only if the format of the field is incorrect or if the field has not been assigned a value. For a definition of the INFO Tag contents see the *Nielsen PCM-to-ID3 SDK Developer’s Guide*.

Figure 5 shows an example of an INFO Tag listing:

```
INFO TAG:
Vendor ID: 999
Device Type: Transcoder or related device
Device ID: 123ABC456DEF
System Version: 123
SDK Version: 2.3
Audio CODEC: PCM 48K
Distributor ID: www.XYZ.com
```

Figure 5 – INFO Tag Contents

Setting Requirements

You must set the following as described:

- **Vendor ID:** set by your application to the value that Nielsen assigned to you.
- **Device ID:** uniquely identify the device on which your application is running. If this is not possible, use the Device ID field to identify the name of the product on which your software is running.

- **System Version:** uniquely identify the current release of your software. Express the version number in any format that you choose, as long as the setting has a one-to-one relationship to your current system version number.
- **SDK Version:** set by PCM-to-ID3 SDK.
- **Distributor ID:** Ensure this field is configurable so that your client (a broadcast or cable facility, for example) can set it when the device is installed.

Test Results

Important A test result of "No status" indicates that a test was not performed. It does **not** indicate that the test failed. Likewise, a setting of "Warn" should prompt you to look closely at the log file for error messages, but it is not an indication that the test failed.

Test 1 Result: DATA Tag Count

Format

Format	**`<count of DATA tags / minimum DATA tags required / 0>`**
Count	Number found in stream
Minimum Required	Minimum required to pass the test
0	Not used

Meaning

Ideally, DATA tags arrive at 10-second intervals. In the presence of many watermarks, however, the tags may arrive more frequently. If this test fails, the stream may contain too few DATA tags.

Example

The example below shows Validator found 310 DATA tags in the stream. The test passed because Validator must find at least 237 DATA tags in the stream.

```
ID3 Tag: DATA Tag Count:           Pass <310/237/0>
```

Possible Causes of Failure

The audio stream fed to the PCM-to-ID3 SDK contained fewer watermarks than expected because of one or more of the following factors:

- The audio is silent.
- The audio is difficult to encode.
- Segments of un-watermarked audio were spliced into the original watermarked content.

- The audio was compressed or resampled since it was first encoded. Compression and resampling can produce audio distortions that negatively affect NAES watermark decoding. Watermarked audio in the range 750 Hz to 6.3 kHz must not be clipped or significantly distorted by compression, resampling, or other processing:
- The PCM-to-ID3 SDK requires that your application maintain a system clock by correctly setting the `currentTime` argument to the `InputAudioData()` function. The `currentTime` argument must advance 1 second for each second of audio content processed. If the value of `currentTime` is not correct, the timing of DATA tags is also incorrect.

Test 2 Result: Tag Format Test

Format

Format	**`<number of tag-format errors / 0 / maximum tolerated>`**
Number of Errors	Number of errors detected
0	Not used
Maximum Tolerated	Maximum number of errors tolerated before failure is declared

Meaning

Validator inspects the placement of the first three forward slash (“/”) characters after the *www.nielsen.com* string in the Nielsen ID3 tag. It also verifies the correct placement of the last characters in the tag. This second check helps determine the presence of both transport-packet payloads that comprise a single ID3 tag. If specific tag fields are missing or are set improperly, Validator declares a tag-format error.

When Validator is run with the -d, -df, or -dr option (used with MPEG-DASH), Validator confirms that the full 20 bytes of ID3 tag headers and the final two ID3 tag characters (null character, followed by the ASCII character *A*) are included in the emsg box. In other words, all 271 bytes of the ID3 tag must be included, unaltered, in the emsg box. If Validator detects that any tags were shortened or altered in any way, Validator logs a tag-format error.

Example

In the Periodic and Summary Reports, the tag-format check compares the number of tag-format errors to the maximum number of errors permitted for the period. The example below shows Validator found no errors. The test passed because Validator allows up to 32 errors.

```
ID3 Tag: Tag Format Test:          Pass <0/0/32>
```

Possible Causes of Failure

The corresponding messages in the log file indicate occurrences of tag format errors with a code of -51, -30, or -3, as shown below; the messages provide insight into the cause of the errors.

```
-51: Corrupted tag (slash error): Sequence number xx, tag time xx.
-51: Corrupted tag (checksum error): Sequence number xx, tag time
xx.
-51: Tag error: Error decoding ID3 tag.
-51: Tag Error: Detected corrupted, partial, or non-Nielsen tag.
-51: Tag error: Separators missing.
-3: Emsg Error: Missing private-data byte at end of frame.
-30: Emsg Error: Missing ID3 Tag headers. Reported up to 5 times
only.
```

Test 3 Result: Sequence Number Test

Format

Format	`<number of sequence-number errors / 0 / maximum tolerated>`
Number of Errors	Number of errors detected
0	Not used
Maximum Tolerated	Maximum number of errors tolerated before failure is declared

Meaning

The sequence-number check indicates the number of times that Validator detected a sequence number was out of order. For example, if Validator detected that tag 102 followed tag 100, Validator reports a sequence number error because tag 101 was missing.

Note that INFO tags sometimes occur out of order, especially at the beginning of the stream. As long as INFO tags are present in the stream, it is acceptable for INFO tags to fall slightly out of order. Ignore any sequence-number test failures that result from out-of-order INFO tags.

Example

The example below Validator found four missing sequence numbers. The test passed because Validator allows up to six errors.

```
ID3 Tag: Sequence Number Test:     Pass <4/0/6>
```

Possible Causes of Failure

The corresponding messages in the log file indicate occurrences of missing sequence numbers with a code of -52 as shown below:

```
-52: Seqno Error, No Time Gap: Sequence number xx, tag time xx.
```

```
-52: Seqno Error and Gap in Tag Times (xx seconds): Sequence
Number xx, Tag time xx.
```

The PCM-to-ID3 SDK issues a new sequence number for each ID3 tag (INFO or DATA) that is generated. A missing sequence number indicates that an ID3 tag with that sequence number was generated, but the tag was not delivered to the ID3 tag stream being evaluated.

The reason for dropped tags is related to the type of file being evaluated. If a viewer fast-forwards during playback on a tablet, the log file may indicate ID3 tags were dropped.

If the tags were extracted from an MPEG-2 transport stream or an MPEG-DASH stream, the multiplexer or packager could have failed to include the ID3 tags in the PES. In investigating tag-sequence errors, you must consider all possible points along the distribution chain where tags potentially could be dropped.

In a facility with a fail-over system, the tag sequence numbers of the primary and fail-over system are not synchronized. The tag times and watermark times of the two systems, however, should be close. A tag overlap or a tag gap of up to 9 seconds may exist at the point of the fail-over. If there are sequence number errors, but no tag-gap errors, ignore the sequence-number errors because you can safely assume they result from switching to a fail-over system.

Test 4 Result: Time Gap Test

Format

Format	**`<number of time-gap errors / 0 / maximum tolerated>`**
Number of Errors	Number of errors detected
0	Not used
Maximum Tolerated	Maximum number of errors tolerated before failure is declared

Test Result Meaning

Under ideal conditions, each DATA tag begins when the previous tag ends. A notable exception to this rule occurs when a tag stream includes ID3 tags from two different tag-generators (a primary device and a fail-over device). If the tag start and stop times indicate that a time gap or time overlap of more than 9 seconds occurred between tags, Validator reports a time-gap error. Validator ignores any gap or overlap of 9 seconds or fewer. See also “Test 13: INFO Tag Interval Test” on page 33.

Example

The example below shows that Validator found three gaps in time between consecutive tags. The test passed because Validator allows up to six gaps.

```
ID3 Tag: Time Gap Test:            Pass <3/0/6>
```

Possible Causes of Failure

The following log messages may appear if Validator detects a time-gap between tags or if tags “overlap” previous tags:

```
-52: Seqno Error and Gap in Tag Times (xx seconds): Sequence
Number xx, Tag time xx.
-74: Tag Time Error, Sequence Number xx: Tag overlaps previous
tag.
```

- If a section of the audio stream does not contain watermarks for 10 seconds or longer, the PCM-to-ID3 SDK does not generate an ID3 tag for that segment. A time gap between consecutive tags would exist. The sequence numbers would be consecutive, however, which indicates that no generated tags were dropped.
- If ID3 tags are dropped in transmission or during the multiplexing process, a sequence-number error accompanies the time-gap error.

Test 5 Result: Tag Duration Test

Format

Format	**`<number of tag duration errors / 0 / maximum tolerated>`**
Number of Errors	Number of errors detected
0	Not used
Maximum Tolerated	Maximum number of errors tolerated before failure is declared

Meaning

Under ideal conditions, a DATA tag spans 10 seconds. If 10 watermarks arrive in less than 10 seconds, the tag-duration is shorter. A tag that is shorter than 10 seconds is not considered faulty.

Regardless of the number of watermarks or EDUs, Validator reports a tag-duration error for any tag that is shorter than 4 seconds or longer than 12 seconds. For a description of EDUs, see “EDUs (Elemental Data Units)” on page 46.

Example

The example below shows Validator found two tags that were shorter than 4 seconds or longer than 12 seconds. The test passed because Validator allows up to seven such tag-duration errors.

```
ID3 Tag: Tag Duration Test:        Pass <2/0/7>
```

Possible Causes of Failure

The following log messages indicate that a tag was too short or too long:

```
-54: Tag Time Error, Sequence Number xx: Tag is short but EDU not
filled.
-54: Tag Time Error, Sequence Number xx: Tag duration is less than
4 seconds.
-54: Tag Time Error, Sequence Number xx: Tag duration is greater
than 12 seconds.
```

The PCM-to-ID3 SDK requires that the calling application maintain a system clock by correctly setting the `currentTime` argument to the `InputAudioData()` function. This setting must advance by 1 second for each second of audio content processed. An incorrect `currentTime` value can result in a tag-duration error.

Test 6 Result: CID SID Test

Format

Format	**`<number of CID SID errors / 0 / maximum tolerated>`**
Number of Errors	Number of errors detected
0	Not used
Maximum Tolerated	Maximum number of errors tolerated before failure is declared

Meaning

Each ID3 tag includes a PC CID (PC Content ID) and an FD CID. The CID has two components: a Nielsen SID and a Nielsen timestamp, both derived from recent Nielsen watermarks. Consequently, the following must occur:

- If the ID3 tag includes valid PC watermarks, the PC CID SID must be set to the value of the PC watermark SID.
- If the ID3 tag includes valid FD watermarks, the FD CID SID must be set to the value of the FD watermark SID.

If the corresponding CID SID value is not set correctly, a CID SID error is declared.

Important   If the watermark SID changes during PCM-to-ID3 SDK processing, it may take up to 20 seconds for the corresponding CID SID to reflect the change. During this transition period, the CID SID may be set to 0, or it may retain the value of the previous SID. Either case is reported as an error, but ignore such errors.

For any certification tests, a failure of the CID SID test is not unusual. If most of the CID SID errors listed in the log file or noted in the Comma-Delimited Tag Listing are associated with the watermark SID transitions that periodically occur, ignore the errors.

Example

The example below shows that there were 33 instances of CID SID errors. Because only 32 errors are tolerated, the test failed.

```
ID3 Tag: CID SID Test:              Fail <33/0/32>
```

Possible Causes of Failure

The most common log message associated with a CID SID error is:

```
-55: PC CID SID error, Sequence Number xx, Tag time xx.
-55: FD CID SID error, Sequence Number xx, Tag time xx.
```

If any of the following events triggers a CID SID error, ignore the error:

- One or more tags contain watermarks with SIDs that are changing.
- More than 10 seconds have transpired without any watermarks. In this case, the CID SID reverts to 0 as the PCM-to-ID3 SDK aligns to the next group of watermarks.
- The watermark decoder may not have properly decoded the SID in the source watermark possibly because the audio is difficult to decode due to compression or improper resampling. The issue produces EDU errors, which results in CID errors.

Test 7 Result: CID Timestamp Test

Format

Format	**`<number of CID timestamp errors / 0 / maximum tolerated>`**
Number of Errors	Number of errors detected
0	Not used
Maximum Tolerated	Maximum number of errors tolerated before failure is declared

Meaning

Important The PCM-to-ID 3 SDK currently only supports Streaming Mode = 0. Streaming Modes 1 and 2 are not currently supported.

For content that was originally watermarked as live, broadcast, streaming content, set the CID timestamp to 3 a.m. If the value is not 3 a.m., an error is declared. The only case in which the CID timestamp should not be set to 3 a.m. is when the watermarks in the tags have TICs and the TICs are set to a value less than 10,800 (1/1/1995, 3 a.m.). See the Glossary for a definition of TICs.

Example

The example below shows that all CID timestamps were properly set. The test passed because Validator allows up to 32 CID timestamp errors.

```
ID3 Tag: CID Timestamp Test:       Pass <0/0/32>
```

Possible Causes of Failure

The most common log messages corresponding to CID timestamp errors are:

```
-55: PC CID TS error, Sequence Number xx, Tag time xx.
-55: FD CID TS error, Sequence Number xx, Tag time xx.
```

The most likely cause of a CID timestamp error is that the PCM-to-ID3 SDK streaming mode was set to 1 or 2 and neither is supported. Set the Streaming Mode to 0.

A timestamp error may also occur if PCM-to-ID3 SDK incorrectly assessed the daylight saving time setting of the NAES watermark. In such cases, the CID timestamp may be set to a value other than 3 a.m. Such cases are truly errors. If you encounter such errors, work with your Nielsen support team to resolve them.

Test 8 Result: CID Timestamp (TS) Offset Test

Format

Format	**`<number of CID timestamp offset errors / 0 / maximum tolerated>`**
Number of Errors	Number of errors detected
0	Not used
Maximum Tolerated	Maximum number of errors tolerated before failure is declared

Test Result Meaning

Each tag includes an offset field that indicates the amount of time between one of the tag’s EDUs and the CID timestamp. Valid EDUs within the tag should fall within 10 seconds before or after the sum of the CID timestamp plus the tag offset. For example, if the CID timestamp is 3 a.m. and the CID timestamp offset is 3600, Validator should report watermarks in the ID3 tag within 10 seconds of 4 a.m., which is 3 a.m. + 3600 seconds.

Example

The example below shows that all CID timestamp offsets were properly set. The test passed because Validator allows up to 32 CID timestamp-offset errors.

```
ID3 Tag: CID TS Offset Test:       Pass <1/0/32>
```

Possible Causes of Failure

These are the most common log messages when Validator finds a CID timestamp-offset error:

```
-55: Non-zero tag offset when PC CID TS is 0, at Seqno xx, Tag
Time xx.
-55: Non-zero tag offset when FD CID TS is 0, at Seqno xx, Tag
Time xx.
-55: Non-zero tag offsets when PC and FD CID TS are 0, at Seqno
xx, Tag time xx.
```

Validator may find CID timestamp-offset errors for these reasons:

- Although most locations in North America observe Daylight Saving Time (DST), some locations do not. If the PCM-to-ID3 SDK was running in a time zone with a policy for observing DST that is not the same as that of the NAES watermark encoder that generated the incoming watermarks, Validator may incorrectly report CID TS or CID offset errors during the one-hour transition from DST to standard time. Unless the faulty error reporting continues beyond 5 minutes after the transition, ignore all reported errors during this 1-hour transition period.

- Due to decoding errors, the tag contains watermarks with invalid time codes. The CID timestamp offset value is calculated based on watermark timestamps. In fact, the number of CID timestamp offset errors directly relates to the number of watermark time warnings. Occasional decoding errors are to be expected and, after careful examination of the log file and the ID3 tag listing, will be excused in most cases. (See “Test 10 Result: Watermark Time Test” on page 30.)
- The PCM-to-ID3 SDK may have reset the offset to 0 because watermark SIDs are in transition. Although reported as an error, a setting of 0, in this case, is acceptable, if it lasts for only one or two tags (10–20 seconds).

Test 9 Result: Breakout Type Test

Format

Format	**`<number of breakout type errors / 0 / maximum tolerated>`**
Number of Errors	Number of errors detected
0	Not used
Maximum Tolerated	Maximum number of errors tolerated before failure is declared

Meaning

If the breakout type falls outside the range 00–99, inclusive, Validator reports an error. It logs an informational message if the breakout type changes within the ID3 tag stream, but does not declare such a change as an error.

Example

The example below shows Validator found no breakout errors. The test passed because Validator allows up to one error.

```
ID3 Tag: Breakout Type Test:       Pass <0/0/1>
```

Possible Causes of Failure

One of the following log messages appears if the breakout code is out of range or if it changes during processing:

```
-58: Invalid breakout code: Sequence Number xx, Tag Time xx.
-58: Breakout type changed from %d to %d:  Sequence Number xx, Tag
Time xx.
```

The breakout type is a user-configurable value. Your application, however, should include safeguards that guarantee that the breakout code is set to 0.

Test 10 Result: Watermark Time Test

Format

Format	**`<number of watermark time errors / 0 / maximum tolerated>`**
Number of Errors	Number of errors detected
0	Not used
Maximum Tolerated	Maximum number of errors tolerated before failure is declared

Meaning

The PCM-to-ID3 SDK keeps track of any watermarks that have timestamps that appear to jump forward or backward (or remain the same) at a pace that is not synchronized with the running transport-stream or ID3 tag “clock.” It also keeps track of changes in daylight saving time (DST) flags. Any discrepancies between the expected watermark time (or DST flag) and the actual setting are tallied as watermark time warnings.

These observations regarding watermarks are counted as *warnings*, not errors, because ID3 tag-inserter applications are not responsible for *generating* watermarks. Tag-inserter applications simply *report* watermarks and use them to set CID values.

Test 10 provides supporting information if there are reported CID errors in a specific ID3 tag. The watermark warnings should be considered informational only.

Example

The example below shows Validator found 36 time stamp errors. The test passed because Validator allows up to 59 errors.

```
Audio Watermark: Time Code Test:   Pass <36/0/59>
```

Corresponding Log Message

When a watermark time code error occurs, the following message appears in the log file:

```
-59: Warning, Sequence Number xx, EDU xx: Watermark timecode
error.
```

Diagnostic Tips

Watermark issues can result from poor encoding or decoding. If you are running the live test, avoid quiet program content such as talk shows. If the decoder is a problem, you may need to investigate your pre-processing of the audio stream.

Test 11 Result: Watermark Level Test

Format

Format	**`<number of watermark level errors / 0 / maximum tolerated>`**
Number of Errors	Number of errors detected
0	Not used
Maximum Tolerated	Maximum number of errors tolerated before failure is declared

Meaning

Each watermark is labeled with a specific SID and a specific level. The level is defined as Program Content (PC) or some variation of Final Distributor (FD, FD1, FD2, or FD3). The level of any given SID should not change; a PC SID should never become an FD SID, and vice-versa.

Validator declares a warning if the level associated with a SID changes. Note that a level change from PC → FD, followed by a change from FD → PC, may be counted as two warnings, although it may indicate only one error. If test 11 fails, take this double counting into consideration

Currently, Validator declares an error if the level changes to or from “UK” (unknown). You can probably dismiss such errors because they are often the result of faulty or incomplete decoding. There should not, however, be an abundance of “UK” watermarks.

Example

The example below shows Validator found three level changes. The test passed because Validator allows up to 36 level changes.

```
Audio Watermark: Level Test:       Pass <3/0/36>
```

Corresponding Log Message

The following message should appear in the log file when a watermark level error occurs or when the level corresponding to a specific SID changes:

```
-60: Warning, Sequence Number xx, EDU xx: EDU level changed.
```

Test 12 Result: INFO Tag Count

Format

Summary Report

Format	**`<count of INFO tags/ minimum required / 0>`**
Count	Number of INFO tags found in the stream
Minimum Required	Minimum number of INFO tags required
0	Not used

Periodic Report

Format	**`<number of seconds since last INFO tag / 0 / maximum interval between tags>`**
Number of Seconds	Number of seconds since last INFO tag
0	Not used
Maximum Interval	Maximum allowable interval between INFO tags

Meaning

Summary Report

The INFO-tag check reports the number of INFO tags that occurred during the entire length of processed content, as compared to the minimum number of INFO tags that must be seen for the test to pass.

Periodic Reports

The INFO-tag test reports the number of seconds since the last INFO tag was detected in the stream. If Validator has seen no INFO tags within the first 300 seconds of content, the INFO-tag test reports the number of seconds since the start of content.

INFO tags are expected to arrive at 5-minute intervals. If 310 seconds expire without an INFO tag, the INFO-tag check fails for that period.

Examples

Summary Report

In the example below, Validator found nine INFO tags in the stream. The test passed because Validator requires a minimum of seven INFO tags.

```
ID3 Tag: INFO Tag Count:           Pass <9/7/0>
```

Periodic Report

The example below indicates that 10 seconds have passed since the last INFO tag. The test passed because 310 seconds had not transpired.

```
INFO Tag: Elapsed Time Report: Pass (10)
```

Possible Causes of Failure

Test 12 most often fails when the tag-inserter application incorrectly invokes a call to the PCM-to-ID3 SDK function `InputAudioData()`. If you are the tag-inserter application developer, check these two conditions:

- If the PCM-to-ID3 SDK releases a DATA tag and an INFO tag immediately after your application invokes a call to `InputAudioData()`, your application may be dropping the INFO tag. Make certain to transmit all ID3 tags that are generated after `InputAudioData()` returns.
- The PCM-to-ID3 SDK requires that the calling application maintain a system clock by correctly setting the `currentTime` argument to the `InputAudioData()` function. The `currentTime` value must advance 1 second for each second of audio content processed. If the value of `currentTime` is not correct, the timing of INFO and DATA tags is incorrect. If your tag-inserter application is processing file input, make certain that you base your `currentTime` value on the number of seconds of processed data, not on a running system clock because the input file is processed faster than real-time.
- For fail-over protection, there may be two tag-inserter applications in a single broadcast facility, both of which contribute ID3 tags to a single ID3 tag metadata stream. If the switch between devices occurs precisely when an INFO tag is about to be delivered, this configuration of tag-inserter devices may result in missed INFO tags.

Test 13: INFO Tag Interval Test

Format

Format	**`<number of INFO-tag interval errors / 0 / maximum tolerated>`**
Number of Errors	Number of errors detected
0	Not used
Maximum Tolerated	Maximum number of errors tolerated before failure is declared

Meaning

The INFO Tag Interval test reports the number of times an INFO tag appeared too soon or too late, in comparison to the previous INFO tag. Validator may report a tag as missing at 310 seconds and may report the next tag as arriving too soon. As a result, Validator reports two errors instead of just one. Take into account such double reporting when evaluating Test 4 failures. See “Test 4 Result: Time Gap Test” on page 24.

Example

The example below shows Validator found no INFO-tag-interval errors. The test passed because Validator allows up to one error.

```
ID3 Tag: INFO Tag Interval Test:   Pass <0/0/1>
```

Possible Causes of Failure

If INFO tags arrive too soon or too late, your application logs the following error message:

```
-53: INFO tag is missing or late: sequence number xx
```

See “Possible Causes of Failure” for Test 12 on page 33.

Tests 14 and 15: INFO TAG CID and Offset Tests

Format

Format	`<number of INFO Tag CID errors / 0 / maximum tolerated>`
Number of Errors	Number of errors detected
0	Not used
Maximum Tolerated	Maximum number of errors tolerated before failure is declared

Meaning

For INFO tags, the PC and FD CIDs should both be set to the null CID, which is X100zdCIGeIlgZnkYj6UvQ==. If either of the CIDs is set to a value other than the null CID, Validator declares an INFO tag CID error.

Likewise, the tag offsets (PC and FD) for all INFO tags should be set to 0. If any INFO-tag offsets are not set to 0, Validator declares an error.

Example

The example below shows Validator found no INFO-tag CID errors. The test passed because Validator allows one such error. INFO-tag offset errors are reported in the same way.

```
INFO Tag: INFO Tag CID Test:       Pass <0/0/1>
INFO Tag: INFO Tag Offset Test:    Pass <0/0/1>
```

Corresponding Log Message

These log messages correspond to INFO tag CID and INFO tag offset errors:

```
-61: INFO Tag CID error at sequence number xx, time xx
-61: INFO Tag Offset error at sequence number xx, time xx
```

If you see either of these errors, call your Nielsen support representative.

Test 16 through 21 Results: INFO Tag Tests

Format

Format	`<number of INFO Tag errors / 0 / maximum tolerated>`
Number of Errors	Number of errors detected
0	Not used
Maximum Tolerated	Maximum number of errors tolerated before failure is declared

Meaning

Validator keeps track of errors that occur in six of the seven INFO-tag fields.

- Vendor ID
- Device Type
- Device ID
- System Version
- SDK Version
- Distributor ID: This is the only field for which a specific string is required. In most cases, it should contain “www” to indicate that the field is a URL. Make Distributor ID a configurable value with a default setting that represents the vendor (www.vendor_company_name.com, for example). When the device is installed in the field, the Distributor ID should be reset to represent the client’s organization.
- Audio CODEC (no longer checked)

If a setting is missing or invalid, Validator declares an error. The *PCM-to-ID3 SDK Developer’s Guide* identifies the length and format of each field. Because all INFO tags are identical, the SDK does not deliver a message for each observed error.

In most cases, Validator simply verifies that the field is filled and that it does not have any characters considered “illegal” for that field.

Corresponding Log Message

All INFO-tag fields appear in the INFO Tag Listing of the Summary Report. Any incorrectly set fields are clearly marked in the listing. No corresponding messages appear in the log file.

Examples

Each of the lines in Figure 6 shows that no INFO-tag field errors occurred. For each field, only one error would be permitted before the test would be considered to have failed. Because all INFO tags in the stream should have the same settings, if one INFO tag fails, all INFO tags are likely to fail.

```
INFO Tag: Vendor ID Test:          Pass <0/0/1>
INFO Tag: Device Type Test:        Pass <0/0/1>
INFO Tag: Device ID Test:          Pass <0/0/1>
INFO Tag: System Version Test:     Pass <0/0/1>
INFO Tag: SDK Version Test:        Pass <0/0/1>
INFO Tag: Distributor ID Test:     Pass <0/0/1>
```

Figure 6 – INFO Tag Tests

Test 22 Result: Tag Placement Check

Tests 22 through 30 are MPEG-2 transport-stream checks. Validator runs the tests only if the content being evaluated is a transport stream (either a single transport-stream file or a segmented HLS stream).

Format

Format	**`<number of tag-placement errors / 0 / maximum tolerated>`**
Number of Errors	Number of errors detected
0	Not used
Maximum Tolerated	Maximum number of errors tolerated before failure is declared

Meaning

If the incoming content is a transport stream that includes a watermarked AC-3 elementary stream, the tag-placement error field keeps track of the number of times that an ID3 tag appears to be out of sync with the audio content that it represents.

Note If the incoming stream is not a transport stream or the test is not conducted, the status is set to “No Status.”

Corresponding Log Message

If Validator determines that a tag is badly placed in the transport stream, it issues this log message:

```
-64: Tag is xx seconds separated from audio watermarks at sequence
number xx, tag time xx.
```

Example

The example below indicates Validator found no tag-placement errors. The test passed because Validator allows up to seven errors.

```
MPEG-2: Tag Placement Test:        Pass <0/0/7>
```

Possible Causes of Failure

The PCM-to-ID3 SDK requires that the calling application maintain a system clock by correctly setting the `currentTime` argument to the `InputAudioData()` function. The `currentTime` value must advance 1 second for each second of audio content processed. If the value of `currentTime` is not correct, the tag times eventually fall out of sync with the metadata PTS.

Test 23 Result: Tag-Time Test

Format

Format	**`< number of times the PTS and tag times are out-of-sync / 0 / maximum tolerated>`**
Number of Errors	Number of errors detected
0	Not used
Maximum Tolerated	Maximum number of errors tolerated before failure is declared

Meaning

PTS (Presentation Timestamp)

The tag-time test compares the rate of advance of the metadata Presentation Timestamp (PTS) to the rate of advance of the first or last tag time. The PTS is the MPEG-2 feature that synchronizes the component elementary streams of the elementary stream.

The transcoder application must assign to the metadata PES packet the PTS of the audio packet from which the most recent watermark was decoded. The _tags.txt file reports the PTS of the PES (converted to seconds) from which the

ID3 tag was extracted.

Note This test runs only if the input is an MPEG-2 transport or HLS stream; otherwise, the result is set to “No Status.”

Examples

The example below shows Validator found no tag-placement errors. The test passed because Validator allows up to five such errors.

```
MPEG - 2 : Tag Time Test:            Pass <0/0/5>
```

Possible Causes of Failure

If Validator determines that the PTS is out-of-sync with the tag time, it delivers this message to the log file:

```
-64: Warning, Sequence Number xx: Tag time out-of-sync with PTS.
Delta = xx.
```

The most likely cause of a tag-time error is poor multiplexing. For proper multiplexing to occur, the PTS of each ID3 DATA Tag PES header must be set to the PTS of the corresponding audio PES header. If the PTS values are properly set, the transport-stream multiplexer should properly place the tags.

Test 24 Result: PES Packet Header Check

Format

Format	`<count of PES packet headers / 0 / minimum required>`
Number of Errors	Count of PES packet headers found in the stream
0	Minimum count required for test to pass
Maximum Tolerated	Not used

Test Result Meaning

Validator counts the number of PES-packet headers that are associated with the Nielsen ID3 tag metadata stream. There should be at least one of these PES packet headers every 10 seconds.

If non-Nielsen ID3 tags are in the metadata PES, the PES packet-header count may be much higher than expected. In that case, the assessment of “Pass” may not be meaningful because it would indicate only that there are some metadata packets in the stream. “Pass” would not indicate that the number of Nielsen PES packets is adequate.

This field is an “event” count, not an error count. Validator does not issue log messages when it finds a Nielsen ID3 tag PES header.

Example

The example below shows that Validator found 319 Nielsen PES headers in the stream. The test passed because Validator requires at least 237 instances.

```
MPEG-2: PES Packet Header Test:    Pass <319/237/0>
```

Possible Causes of Failure

If the PES headers are missing, even though the PMT lists the stream in its elementary loop, the PES stream itself may have been omitted from the transport stream during the multiplexing process.

Test 25 Result: PES Owner-ID Check

Format

Format	`<count of PES owner-IDs / 0 / minimum required>`
Number of Errors	Count of PES owner-IDs found in the stream
0	Minimum count required for test to pass
Maximum Tolerated	Not used

Meaning

When Validator finds the string “www.nielsen.com” near the beginning of a PES packet, it logs the event. The purpose of this test is to ascertain that the ID3 tag metadata elementary stream is, indeed, a *Nielsen* ID3 tag stream. Validator must only find one Nielsen URL to confirm this fact.

Because this is an “event” check and not an error check, Validator does not log a message when it detects a Nielsen owner ID.

Example

The example below shows that Validator found 319 Nielsen owner IDs. In most cases, the total number is equal to the sum of INFO tags and DATA tags. The test passed because only one owner-ID is required to declare this a Nielsen ID3 tag.

```
MPEG-2: Nielsen Owner ID Test:     Pass <319/1/0>
```

Test 26 Result: Metadata PES PTS Check

Format

Format	`<number of PES-PTS errors / 0 / maximum tolerated>`
Number of Errors	Number of errors detected
0	Not used
Maximum Tolerated	Maximum number of errors tolerated before failure is declared

Meaning

For ID3 DATA tag packets to be properly multiplexed into a transport stream, each PES header of the metadata stream must contain a PTS (Presentation Timestamp). Each time the PTS is missing from a Nielsen metadata header, Validator reports an error.

Corresponding Log Message

If Validator determines that the PTS field is missing from the Nielsen ID3 DATA tag PES header, it logs this error message:

```
-67: Missing PTS in Nielsen PES header.
```

Example

The example below shows that Validator found no PTS fields were missing from any DATA-tag PES headers. The test passed because Validator allows up to 16 missing PTS fields.

```
MPEG-2: Missing PES PTS Test:      Pass <0/0/16>
```

Test 27 Result: PES Header Test

Format

Format	`<number of PES-header errors / 0 / maximum tolerated>`
Number of Errors	Number of errors detected
0	Not used
Maximum Tolerated	Maximum number of errors tolerated before failure is declared

Meaning

Each Nielsen metadata PES packet header must have a stream ID of 0xBD and it must include a non-zero PES packet size, as specified in the document “Timed Metadata for HTTP Live Streaming.” See “Related Documents” on page 7.

Corresponding Log Message

If the PES packet stream ID is incorrectly set or if the PES packet size of a Nielsen ID3 PES packet is set to 0, one of the following error messages is logged:

```
-3:  Warning: PES header not valid for ID3 Tag Metadata Packet
-3:  Warning: PES packet length is set to 0!
```

Example

The example below shows Validator found no PES header errors. The test passed because Validator allows up to five errors.

```
MPEG - 2 : PES Header Test:          Pass <0/0/5>
```

Test 28 Result: PES Incomplete Tag Test

Format

Format	`<number of incomplete ID3 tags / 0 / maximum tolerated>`
Number of Incomplete ID3 Tags	Number detected
0	Not used
Maximum Tolerated	Maximum number of errors tolerated before failure is declared

Meaning

If the size of the payload in the Nielsen ID3 metadata stream is not large enough to hold a complete Nielsen ID3 tag, Validator reports an error. An incomplete payload cannot be decoded.

Each Nielsen ID3 tag spans two transport packets. If the size of the payload (as specified in the PES header) is correct, but the second transport packet contributing to the payload is missing, Validator declares an incomplete-tag error.

Corresponding Log Message

If Validator determines that the metadata PES payload is the wrong size or that the second packet in the payload is missing, it logs a message.

```
-65: Invalid tag returned from tag buffer. Wrong size.
```

Example

The example shows that no PES payloads were the wrong size. The test passed because Validator allows up to seven badly sized PES packets.

```
MPEG-2: PES Incomplete Tag Test:   Pass <0/0/7>
```

Tests 29 and 30 Result: PMT Elementary Stream and Descriptor Checks

Format

Format	**`<count of metadata elementary streams and ID3 metadata descriptors found in the stream / 0 / maximum tolerated>`**
Number of Errors	Count of streams and descriptors
0	Minimum count required for test to pass
Maximum Tolerated	Not used

Meaning

For an MPEG-2 decoder to detect a Nielsen ID3 tag elementary stream, the decoder must find an entry for a stream of type 0x15 followed by an MPEG-2 ID3 tag descriptor in the PMT elementary-stream loop. Each time that the decoder finds a stream of type 0x15 in the loop, Validator counts it as a metadata PES. Each time that it finds an ID3 tag descriptor, Validator counts it as an ID3 tag metadata descriptor.

For more information regarding the proper formation of PMT packets for ID3 Tag Metadata, refer to “HTTP Live Streaming Metadata Spec.” See “Related Documents” on page 7.

Because both of these checks are “event” checks, not error checks, no messages are logged when the tallies are incremented.

Example

The example below shows that the proper descriptor and elementary stream listing were included in hundreds of PMTs. The test passed because only five instances are required.

```
MPEG-2: PMT Descriptor Test:       Pass <667/5/0>
MPEG-2: PMT Elem Stream Test:      Pass <667/5/0>
```

Detailed Tag Listing – DATA Tag

If you set the -p command-line option to 2 or 6, Validator generates a Detailed Tag Listing and reports all fields of each Nielsen ID3 tag that it detects. Remember that there are two types of ID3 tags: DATA and INFO. Ideally, DATA tags are delivered at 10-second intervals and INFO tags are delivered at 5-minute intervals. DATA tags are used for crediting, while INFO tags are used to identify your device or application as the source of the DATA tags.

The text below shows a typical Nielsen ID3 DATA tag. The sections that follow describe the components of the DATA tag. Note the following:

- If you called Validator with the **-df** or **-dr** option, the tag-listing file displays the name of each segment, followed by the string *** EMSG: www.nielsen.com:id3:v1, before it lists the ID3 tags found in that segment.
- If you called Validator with the **-d** option, the *** EMSG string is displayed, but the segment names are not displayed.

```
Tag Number: 7.      Transport Pid: 259 (0x103) PTS (seconds): 65
PC CID: Ma0O2J3MrcrfObLJMbVUYQ== (9020/2800851248, 09/15/2015
03:00:00)
FD CID: a0di85LJo6G-hzfGTaCqZw== (9120/2800851248, 09/15/2015
03:00:00)
Seconds Since FD Day-Start: 57659.
Seconds Since PC Day-Start: 57661.
Break-out Flag: 00
Seq No : 7,  First: 1452506981, Last: 1452506990
Code:   SID, Stacked,     TS/TIC, DST/Brkout, Distribution,
Sample, Offset, time string, time type, Frame Level
 N2 :  9020,       U, 2800908901,     DST,    PC,  5225, 1,
09/15/2015 19:00:53,      TIME
 N6 :  9120,       S,  180039654,     DST,    FD1,  581, 2,
09/15/2015 19:00:54,     TIME
 N2 :  9120,       U, 2800908903,     DST,    FD,  5411, 3,
09/15/2015 19:00:55,    TIME
 N6 :  9020,       U,  180039657,     DST,    PC,  284,  5,
09/15/2015 19:00:57,     TIME
 N2 : 61721,       S, 3845624704,     DST,    UK,  5583, 5,
10/23/2048 09:50:56,    TIME
 N2 :  9020,       U, 2800908905,     DST,    PC,  5599, 5,
09/15/2015 19:00:57,    TIME
 N6 :  9120,       U,  180039658,     DST,    FD1, 582,  6,
09/15/2015 19:00:58,     TIME
 N2 :  9120,       U, 2800908907,     DST,    FD,  5785, 7,
09/15/2015 19:00:59,    TIME
 N2 :  9020,       U, 2800908909,     DST,    PC,  5972, 9,
09/15/2015 19:01:01,    TIME
 N2 :  9020,       S, 2800908909,     DST,    PC, 973, 9,
09/15/2015 19:01:01,      TIME
```

Tag Number

The tag number indicates the order in which Validator detected the tag in the metadata stream. This field is informational only; the tag number is not part of the ID3 tag.

Transport PID

The Transport PID uniquely identifies the metadata PES (packetized elementary stream) that carries the Nielsen ID3 tags. Its setting is read from the MPEG-2 Program Map Table (PMT).

PTS

The PTS field reports the presentation timestamp of the metadata PES packet in which the Nielsen ID3 tag was found. The PTS value, which is truncated to the nearest second, should match the PTS of the last audio packet presented to the PCM-to-ID3 SDK before the ID3 tag was packaged and released.

If this value is not correctly set, the Nielsen ID3 tags may fall out of sync with the audio stream, resulting in improper crediting. The PTS should advance at roughly the same rate as the First and Last tag times.

PC CID

The PC CID is a 24-byte string that holds two encrypted values (SID and timestamp).

Under ideal conditions, the PC CID SID identifies the SID of the last PC watermark listed among the EDUs of the current ID3 tag. The PCM-to-ID3 SDK requires 10 seconds or more, however, to “latch onto” the proper PC SID. As a result, the PC CID may be inaccurate (or set to “X100zdCIGeIlgZnkYj6UvQ”) for 10 to 20 seconds after a SID changes or disappears.

Under ideal conditions, the PC CID timestamp is a raw NAES 2 timestamp that translates to the start of the current broadcast day (3 a.m. in the time zone where the NAES encoder was operating). If the PC CID SID is set to 0 or is inaccurate, crediting ignores the PC CID timestamp.

All CID timestamps are derived from the watermarks to which the PC CID SID corresponds. They are not derived from the “current time” that your application supplies to the PCM-to-ID3 SDK.

FD CID

The FD CID is a 24-byte string that holds two encrypted values (SID and timestamp).

Under ideal conditions, the FD CID SID identifies the SID of the last NAES 2 FD watermark listed among the EDUs of the current ID3 tag. In the absence of NAES 2 watermarks, special rules apply to the selection of the SID based on NAES 6 watermarks. The PCM-to-ID3 SDK requires 10 seconds or more to “latch onto” the proper FD SID. As a result, the FD CID may be inaccurate (or set to 0) for 10 to 20 seconds after a SID changes or disappears.

Under ideal conditions, the FD CID timestamp is a raw NAES 2 timestamp that translates to the start of the current broadcast day (3 a.m. in the time zone where the NAES encoder was operating). If the FD CID SID is set to 0 or is inaccurate, crediting ignores the FD CID timestamp.

All CID timestamps are derived from the watermarks to which the PC CID SID corresponds. They are not derived from the “current time” that your application supplies to the PCM-to-ID3 SDK.

Seconds since FD Day-Start (FD CID Offset)

The FD CID offset specifies the number of seconds that separate the FD CID timestamp and the timestamp of the most recent EDU that bears the FD CID SID. Note that all timestamps are translated to NAES 2 format before any calculations are made. The following lines show an ideal example:

```
(<FD CID timestamp> + <FD CID Offset> - 15) <= FD EDU timestamp <=
(<FD CID timestamp> + <FD CID Offset> + 15)
```

Seconds since PC Day-Start (PC CID Offset)

The PC CID offset specifies the number of seconds that separate the PC CID timestamp and the timestamp of the most recent EDU that bears the PC CID SID. Note that all timestamps are translated to NAES 2 format before any calculations are made. The following lines show an ideal example:

```
(<PC CID timestamp> + <PC CID Offset> - 15) <= PC EDU timestamp <=
(<PC CID timestamp> + <PC CID Offset> + 15)
```

Break-out Flag

Always set the breakout code to 0 in your application.

Sequence Number (Seqno)

The PCM-to-ID3 SDK generates a unique sequence number for each DATA or INFO tag that it creates. Sequence numbers are consecutive, and INFO tags are numbered in sequence with DATA tags. A missing sequence number in the detailed tag listing is usually an indication that the ID3 tag was dropped or delivered out of order sometime after the PCM-to-ID3 SDK generated it. Out-of-order delivery of INFO tags is acceptable. DATA tags, however, should appear in the output stream in the same order as the PCM-to-ID3 SDK generated them.

Tag Times (First, Last)

Each ID3 tag includes a First and Last tag time. These values are derived from the “current time” setting that your application feeds to the PCM-to-ID3 SDK with each call to `InputAudioData()`.

- The First tag time identifies the time when the current ID3 tag began to collect incoming watermarks.
- The Last tag time identifies the time when the ID3 tag was packaged and delivered to your application.

On average, Nielsen ID3 tags should be 10 seconds long. If the PCM-to-ID3 SDK receives 10 or more EDUs in less than 10 seconds, however, the tag ends promptly after the tenth EDU is decoded.

Ideally, the Last tag time of the previous DATA tag equals the First tag time of the current DATA tag. If there is a gap between tag times, combined with a missing sequence number, we can be relatively certain that a tag that was generated by the PCM-to-ID3 SDK was dropped. If there is a gap between tag times, but no sequence number is missing, it is likely that the PCM-to-ID3 SDK was presented a segment of un-watermarked audio data. In the absence of watermarks, no ID3 tags are generated.

EDUs (Elemental Data Units)

The PCM-to-ID3 SDK receives an audio stream from your application, summarizes the contents of the watermark by creating an EDU, and stores the EDU in an ID3 tag. This means that there is a “watermark SID” (the SID in the source watermark) and the EDU is a representation of that SID. Each ID3 tag may contain up to 10 EDUs.

The following sections give examples and descriptions of some of the typical EDUs. Fields in blue font in the snippets correspond with blue font in the table that follows each example.

NAES 2 EDU

```
Tag Number: 7.   Transport Pid: 259 (0x103) PTS (seconds): 65
PC CID: Ma0O2J3MrcrfObLJMbVUYQ== (9020/2800851248, 09/15/2015 03:00:00)
FD CID: a0di85LJo6G-hzfGTaCqZw== (9120/2800851248, 09/15/2015 03:00:00)
Seconds Since FD Day-Start: 57659.
Seconds Since PC Day-Start: 57661.
Break-out Flag: 00
Seq No : 7,  First: 1452506981, Last: 1452506990
Code:   SID, Stacked,     TS/TIC, DST/Brkout, Distribution,  Sample, Offset,
time string, time type, Frame Level
 N2 :  9020,        U, 2800908901,         DST,            PC,    5225,       1,
09/15/2015 19:00:53,       TIME
 N6 :  9120,         S,  180039654,         DST,           FD1,     581,       2,
09/15/2015 19:00:54,       TIME
```

Table 5 – NAES 2 EDU Description

Field	Value	Description
Code	N2	EDU was derived from NAES 2 watermark
SID	9020	SID for source watermark is 9020. The PC CID is also 9020.
Stacked	U	Watermark is “unstacked,” which means that all reported fields are generally more reliable than those of a stacked (S) watermark
TS/TIC	2800908901	Raw N2 time stamp. 2800908901 translates to the value in the Time String field (see below). In the PC CID snippet, notice that the timestamp is 2800851248. The difference between 2800908901 and 2800851248 is 57653, which is within 10 seconds of Seconds Since PC Day-Start: 57661.
DST/Brkout	DST	NAES encoder generated the watermark during daylight saving time
Distribution	PC	Watermark level is Program Content
Sample	5225	Nielsen admin use only
Offset	1	PCM-to-ID3 SDK decoded this watermark 1 second after the start of this ID3 tag
Time String	09/15/2015 19:00:53	See TS/TIC above.
Time Type	TIME	Timestamp of the EDU (see TS/TIC above). This represents clock time, not time-in-content (TIC).
Frame Level	Not used	

NAES 6 EDU

```
Tag Number: 7.   Transport Pid: 259 (0x103) PTS (seconds): 65
PC CID: Ma0O2J3MrcrfObLJMbVUYQ== (9020/2800851248, 09/15/2015 03:00:00)
FD CID: a0di85LJo6G-hzfGTaCqZw== (9120/2800851248, 09/15/2015 03:00:00)
Seconds Since FD Day-Start: 57659.
Seconds Since PC Day-Start: 57661.
Break-out Flag: 00
Seq No : 7,  First: 1452506981, Last: 1452506990
Code:   SID, Stacked,     TS/TIC, DST/Brkout, Distribution,  Sample, Offset,
time string, time type, Frame Level
 N2 :  9020,        U, 2800908901,        DST,            PC,     5225,      1,
09/15/2015 19:00:53,      TIME
.
.
N6 :  9120,        U,  180039658,        DST,           FD1,      582,      6,
09/15/2015 19:00:58,      TIME
```

Table 6 – NAES 6 EDU Description

Field	Value	Description
Code	N6	EDU was derived from NAES 6 watermark
SID	9120	SID for source watermark is 9120. The FD CID is also 9120.
Stacked	U	Watermark is “unstacked,” which means that all reported fields are generally more reliable than those of a stacked (S) watermark
TS/TIC	180039658	Raw N6 time stamp. 180039658 translates to the value in the Time String field (see below). The NAES 2 equivalent of this timecode is 2800908906. In the snippet above, the FD CID timestamp is 2800851248. The difference between 2800908906 and 2800851248 is 57658, which is within 10 seconds of the value for FD Day-Start: 57659.
DST/Brkout	DST	NAES encoder generated the watermark during daylight saving time
Distribution	FD1	Watermark level is Final Distributor 1
Sample	582	Nielsen admin use only
Offset	6	PCM-to-ID3 SDK decoded this watermark 6 seconds after the start of this ID3 tag
Time string	09/15/2015 19:00:58	See TS/TIC above.
Time type	TIME	Timestamp of the EDU (see TS/TIC above). This represents clock time, not time-in-content (TIC).
Frame level	Not used	

Evaluating EDUs for Inclusion in the Summary SID List

During normal processing, the PCM-to-ID3 SDK may encounter bit-errors when decoding watermarks. As a result, one or more of the fields in the decoded watermark may be incorrect. We expect to see occasional bit errors among the EDUs. However, if the number of erroneous EDUs exceeds the number of valid EDUs, there may be a problem with the quality of the audio being presented to the PCM-to-ID3 SDK.

When processing a stream of ID3 tags, Validator frequently identifies the first few watermarks with a specific SID as having an “undefined” level (marked “UK” in the EDU line). Eventually the level is correctly identified as either PC or FD. When generating the SID list, Validator includes the UK watermarks in the tally of watermarks with the properly identified level.

```
N2 :  9999,       S,  663929984,        STD,           UK,    3058,
2, 01/15/2016 08:59:44,     TIME
N2 :  9999,       S,  663929984,        STD,           UK,    3993,
2, 01/15/2016 08:59:44,
N2 :  9999,       U,  663930087,        STD,           FD,    8107,
6, 01/15/2016 09:01:27,     TIME
N2 :  9999,       U,  663930089,        STD,           FD,    8107,
6, 01/15/2016 09:01:29,     TIME
N2 :  9999,       S,  663929984,        STD,           UK,    3993,
2, 01/15/2016 08:59:44,     TIME
```

Figure 7 – Tags.txt File

In example shown in Figure 9, the count for N2 FD SID 9999 is 5 because the level of the third EDU is FD. The third EDU, as well as all subsequent and previous EDUs with the N2 SID 9999 marked as FD or UK, would be included in the tally of watermarks with SID 9999.

If SID 9999 is misidentified as a PC SID somewhere in the stream, Validator adds a separate line to the SID listing. In the case where a SID is identified as both PC and FD within a single stream, Validator adds the UK tally to whichever of the two entries has the higher count. For example, suppose that Validator found that SID 9999 was represented 5 times as PC and later was identified 100 times as FD. Interspersed among the PC and FD watermarks, there were 20 watermarks with SID 9999 that were identified as UK. In this case, the Summary Report SID listing for SID 9999 would have 2 lines:

```
PC SID 9999: NAES 2 count = 5
FD SID 9999:  NAES 2 count = 120
```

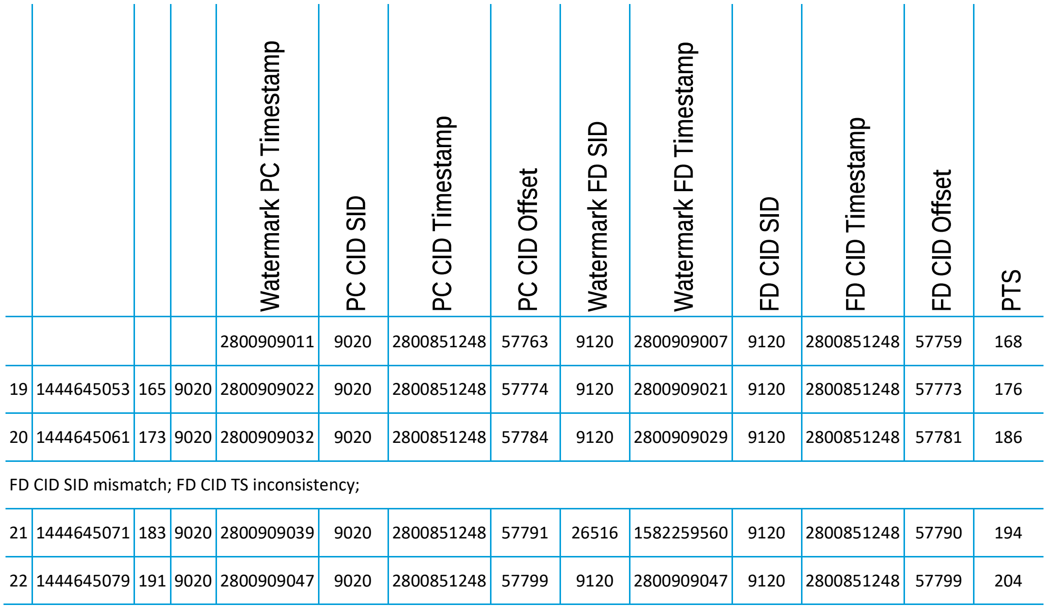
Watermarks with SIDs that are never represented as either PC or FD are not included in the SID listing of the Summary Report.

Comma-Delimited Tag Listing

If you set the -p option to 4 or 6 when you run Validator, Validator generates a Comma-Delimited Tag Listing. To take advantage of this format, import the listing into a spreadsheet.

Example

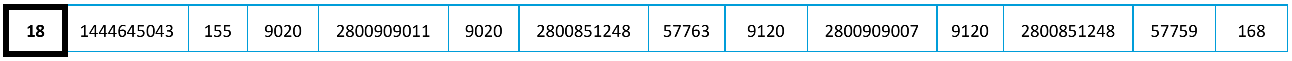
Table 7 –Comma-Delimited Tag Listing Imported into Microsoft Excel with Added Headings



				Watermark PC Timestamp	PC CID SID	PC CID Timestamp	PC CID Offset	Watermark FD SID	Watermark FD Timestamp	FD CID SID	FD CID Timestamp	FD CID Offset	PTS
				2800909011	9020	2800851248	57763	9120	2800909007	9120	2800851248	57759	168
19	1444645053	165	9020	2800909022	9020	2800851248	57774	9120	2800909021	9120	2800851248	57773	176
20	1444645061	173	9020	2800909032	9020	2800851248	57784	9120	2800909029	9120	2800851248	57781	186
FD CID SID mismatch; FD CID TS inconsistency;													
21	1444645071	183	9020	2800909039	9020	2800851248	57791	26516	1582259560	9120	2800851248	57790	194
22	1444645079	191	9020	2800909047	9020	2800851248	57799	9120	2800909047	9120	2800851248	57799	204

Table 7 shows five rows of a Comma-Delimited Tag Listing imported into Microsoft Excel (with heading row and labels added). The following items describe each of the fields.

- **Sequence Number**

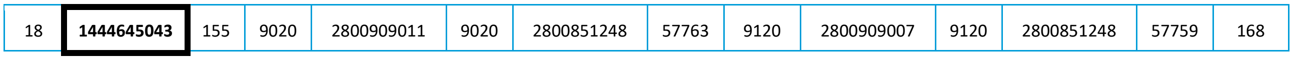


18	1444645043	155	9020	2800909011	9020	2800851248	57763	9120	2800909007	9120	2800851248	57759	168

The PCM-to-ID3 SDK generates a unique sequence number for each tag (DATA or INFO) that it creates. Sequence numbers are consecutive, and INFO tags are numbered in sequence with DATA tags. A missing sequence number in the detailed tag listing usually is an indication that the ID3 tag was dropped or delivered out of order sometime after the PCM-to-ID3 SDK generated it. It is acceptable for INFO Tags to be delivered out of order. However, DATA tags should appear in the output stream in the same order as the PCM-to-ID3 SDK generated them.

Table 7 shows that all five sequence numbers from 18 through 22 are present. The Comma-Delimited Tag Listing, especially in a spreadsheet, allows you to quickly determine if an ID3 tag is missing.

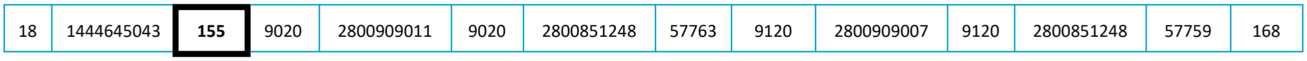
- **Tag Time**



18	**1444645043**	155	9020	2800909011	9020	2800851248	57763	9120	2800909007	9120	2800851248	57759	168

The Tag Time corresponds to the First tag time, which identifies the time at which the PCM-to-ID3 SDK began to collect watermarks for the current Nielsen ID3 tag. The tag times of consecutive DATA rows should be no more than 12 seconds apart, assuming that the processed audio contains uninterrupted Nielsen watermarks. In the snippet above from Table 7, all tag times are correctly spaced.

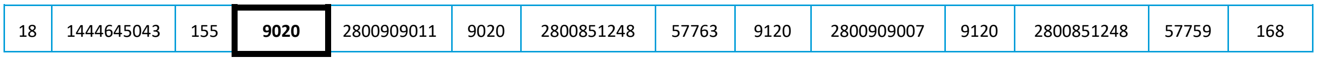
- **Tag Offset**



18	1444645043	**155**	9020	2800909011	9020	2800851248	57763	9120	2800909007	9120	2800851248	57759	168

The tag offset field reports the offset of the current tag from the First tag listed in the file. The offset is calculated by subtracting the First tag time from the current tag time. If no tags are missing, and if the audio is properly watermarked, the tag offsets should be spaced at intervals of 12 seconds or less.

- **Watermark PC SID**

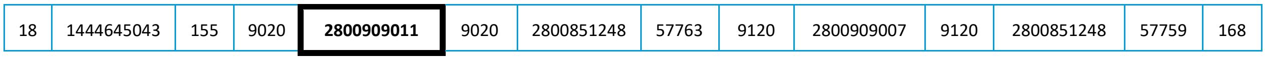


18	1444645043	155	**9020**	2800909011	9020	2800851248	57763	9120	2800909007	9120	2800851248	57759	168

The Watermark PC SID identifies the SID of the last valid PC EDU in the current tag. In a spreadsheet, quickly scan this column to determine whether—and, if so, at what point—the PC SID changed from the start of processing to the end. Around the point of transition, expect to see 10–20 seconds of PC CID errors (SID and timestamp). Because the PC SID value is changing, discount these errors.

Note Each of the PC CID errors is included in the CID SID error count of the Summary Report.

- **Watermark PC Timestamp**



18	1444645043	155	9020	**2800909011**	9020	2800851248	57763	9120	2800909007	9120	2800851248	57759	168

The Watermark PC timestamp identifies the calculated NAES 2 timestamp of the last valid EDU whose SID matches the Watermark PC SID (9020 in the snippet above). If the last valid PC EDU with the corresponding SID is derived from a NAES 6 watermark, the Watermark PC Timestamp is converted from NAES 6 to NAES 2 format; otherwise, the timestamp is directly read from the EDU.

Occasionally decoding errors may cause one or more bits of the timestamp to be incorrectly set. In a spreadsheet, you can easily detect the faulty timestamp.

- **PC CID SID**

18	1444645043	155	9020	2800909011	**9020**	2800851248	57763	9120	2800909007	9120	2800851248	57759	168

The PC CID SID is the decrypted SID component of the 24-byte PC Content ID.

Under ideal conditions, the PC CID SID identifies the SID of the last PC watermark listed among the EDUs of the current ID3 tag. It takes the PCM-to-ID3 SDK 10 seconds or more to “latch onto” the proper PC SID. As a result, the PC CID may be inaccurate (or set to 0) for 10–20 seconds after a SID changes or disappears.

In the snippet above, the PC CID SID may be set to 0 (or to the value of the previous PC SID) for two or three tags after the Watermark PC SID changes to a new value. The “Comments” field usually reports such PC CID SID mismatches. Table 7 shows such a Comment. Ignore them if they fall within three rows of a changing PC SID.

- **PC CID Timestamp**

18	1444645043	155	9020	2800909011	9020	**2800851248**	57763	9120	2800909007	9120	2800851248	57759	168

The PC CID Timestamp is the decrypted timestamp component of the 24-byte PC Content ID.

Under ideal conditions, the PC CID timestamp is a raw NAES 2 timestamp that translates to the start of the current broadcast day (3 a.m. in the time zone where the NAES encoder was operating). If the PC CID SID is set to 0 or is inaccurate, ignore the PC CID timestamp.

All CID timestamps are derived from the watermarks to which the PC CID SID corresponds. They are not derived from the “current time” that your application supplies to the PCM-to-ID3 SDK.

- **PC CID Offset**

18	1444645043	155	9020	**2800909011**	9020	**2800851248**	**57763**	9120	2800909007	9120	2800851248	57759	168

The PC CID offset specifies the number of seconds that separate the PC CID timestamp and a representative Watermark PC Timestamp for the current tag. Note that all timestamps are translated to NAES 2 format before any calculations are made. Ideally:

```
(<PC CID timestamp> + <PC CID Offset> - 10) <= Watermark PC
Timestamp <=
(<PC CID timestamp> + <PC CID Offset> + 10)
```

In the snippet above from the Comma-Delimited Tag Listing in Table 7, note that, for sequence number 18:

- PC CID TS = 2800851248
- PC CID Offset = 57763

Therefore,

- PC CID TS + offset = 2800909011 (calculated)
- Watermark PC timestamp = 2800909011

The watermark PC timestamp exactly matches the CID TS + offset, so the values are correct.

- **Watermark FD SID**

18	1444645043	155	9020	2800909011	9020	2800851248	57763	**9120**	2800909007	9120	2800851248	57759	168

The Watermark FD SID identifies the SID of the last valid FD EDU in the current tag. In a spreadsheet, you can easily scan this column to determine whether the FD SID changed from the start of processing to the end. If the FD SID changed, you can quickly determine whether—and, if so, at what point—the change occurred. Around the point of transition, expect to see 10–20 seconds of FD CID errors (SID and timestamp). Because the FD SID value is changing, discount these errors.

- **Watermark FD Timestamp**

18	1444645043	155	9020	2800909011	9020	2800851248	57763	9120	**2800909007**	9120	2800851248	57759	168

The Watermark FD timestamp identifies the calculated NAES 2 timestamp of the last valid EDU with the SID that matches the Watermark FD SID. If the last valid FD EDU with the corresponding SID is derived from a NAES 6 watermark, the Watermark FD Timestamp is converted from NAES 6 to NAES 2 format; otherwise, the timestamp is directly read from the EDU.

Occasionally decoding errors may cause one or more bits of the timestamp to be incorrectly set. In a spreadsheet, you can easily detect the faulty timestamp.

- **FD CID SID**

20	1444645061	173	9020	2800909032	9020	2800851248	57784	9120	2800909029	**9120**	2800851248	57781	186
FD CID SID mismatch; FD CID TS inconsistency;													
21	1444645071	183	9020	2800909039	9020	2800851248	57791	26516	1582259560	**9120**	2800851248	57790	194

The FD CID SID is the decrypted SID component of the 24-byte FD Content ID.

Under ideal conditions, the FD CID SID identifies the SID of the last FD watermark listed among the EDUs of the current ID3 tag. It takes the PCM-to-ID3 SDK 10 seconds or more to “latch onto” the proper FD SID. As a result, the FD CID may be inaccurate (or set to 0) for 10–20 seconds after a SID changes or disappears.

In the example above, the FD CID SID may be set to 0 (or to the value of the previous FD SID) for two or three tags after the Watermark FD SID changes to a new value. The “Comments” field usually reports such FD CID SID mismatches. Ignore them if they fall within three rows of a changing FD SID. The snippet above shows such a Comment.

The tag with sequence number 21 has a badly decoded FD SID. As a result, the “Comments” field displays “FD CID SID mismatch.” This error should be ignored, since the CID is correct, and only the decoded watermark is in error.

- **FD CID Timestamp**

20	1444645061	173	9020	2800909032	9020	2800851248	57784	9120	2800909029	9120	**2800851248**	57781	186

FD CID SID mismatch; FD CID TS inconsistency													
21	1444645071	183	9020	2800909039	9020	2800851248	57791	26516	1582259560	9120	**2800851248**	57790	194

The FD CID Timestamp is the decrypted timestamp component of the 24-byte PC Content ID.

Under ideal conditions, the FD CID timestamp is a raw NAES 2 timestamp that translates to the start of the current broadcast day (3 a.m. in the time zone where the NAES encoder was operating). If the FD CID SID is set to 0 or is inaccurate, the FD CID timestamp should be ignored.

All CID timestamps are derived from the watermarks to which the FD CID SID corresponds. They are not derived from the “current time” that your application supplies to the PCM-to-ID3 SDK.

The snippet above shows the tag with sequence number 21, which has a badly decoded FD timestamp. As a result, the “Comments” field in this row reports “FD CID TS inconsistency.” Ignore this error because the CID is correct, and only the decoded watermark is in error.

- **FD CID Offset**

18	1444645043	155	9020	2800909011	9020	2800851248	57763	9120	**2800909007**	9120	**2800851248**	57759	168

The FD CID offset specifies the number of seconds that separate the FD CID timestamp and a representative Watermark FD Timestamp for the current tag. Note that all timestamps are translated to NAES 2 format before any calculations are made. Ideally:

```
(<FD CID timestamp> + <FD CID Offset> - 10) <= Watermark FD
Timestamp <=
(<FD CID timestamp> + <FD CID Offset> + 10)
```

In the example comma-delimited file snippet, note that, for sequence number 18:

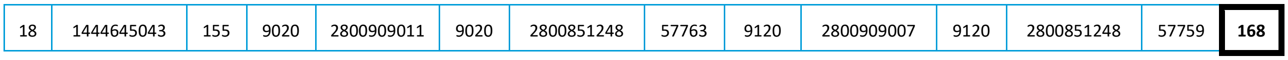
- FD CID TS = 2800851248
- FD CID Offset = 57759

Therefore,

- FD CID TS + offset = 2800909007 (calculated)
- Watermark FD timestamp = 2800909007

The watermark FD timestamp exactly matches the FD CID TS + offset, so the values are correct.

- **PTS**

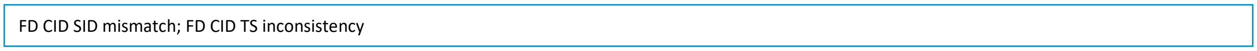


18	1444645043	155	9020	2800909011	9020	2800851248	57763	9120	2800909007	9120	2800851248	57759	**168**

The PTS field reports the presentation timestamp of the metadata PES packet in which the Nielsen ID3 tag was found. The PTS value, which is truncated to the nearest second, should match the PTS of the last audio packet presented to the PCM-to-ID3 SDK before the ID3 tag was packaged and released. If this value is not set correctly, the Nielsen ID3 tags may fall out of sync with the audio stream and result in improper crediting.

The PTS should advance at roughly the same rate as the tag time.

- **Comments**



FD CID SID mismatch; FD CID TS inconsistency

The Comments field may include one or more messages, separated by semicolons.

- CID SID and CID TS errors for the two or three tags often follow the appearance of a new SID.
- Such errors may also accompany badly decoded watermarks with incorrect SIDs or timestamps.

If error messages accompany either of these two situations, ignore the errors.

Troubleshooting Tips

Validator is effective in pinpointing common problems faced by developers who use the PCM-to-ID3 SDK. The remainder of this section lists some of the problems that Validator can help to identify. In most cases, suggested remedies follow the problem description.

Tag Time Not Advancing Properly

Background

Each ID3 tag has a start time and an end time. Ideally, the difference between the start and end times should be 10 seconds. If the tag fills with 10 or more EDUs, the tag-duration is legitimately less than 10 seconds.

Sometimes, though, the tag duration is shorter than 10 seconds for no apparent reason. In such cases, the tag time is not properly advancing.

Symptoms

If there is a real problem with tag times not advancing properly, the Summary Report shows that the Tag Duration test failed as shown in the example below:

```
ID3 Tag: Tag Duration Test:        Fail <35/0/2>
```

Possible Causes

When the application feeds data to the PCM-to-ID3 SDK, it calls this function:

```
void InputAudioData(uint8_t *inBuffer, uint32_t nSize, time_t
currentTime);
```

The third argument to the function must be a value that increments by 1 second for every second of delivered content. The PCM-to-ID3 SDK maintains its clock based on this value. If it does not properly advance, the clock is wrong.

Another cause could be content that is bad or fed to the ID3-tag inserter at the wrong rate.

Suggestions for Follow-up

- Check the `currentTime` argument in your call to the PCM-to-ID3 SDK API `InputAudioData().`
- Send to Nielsen 10 minutes of transport-stream for further analysis.

Processing Set to Incorrect Mode (Streaming or MDI)

Background

In the past, the PCM-to-ID3 SDK could be run in three modes. **Currently only Streaming Mode 0 for streaming or broadcast content is supported.** Do not use MDI mode (modes 2 or 3), which were originally designed for VOD-style content (file or offline).

You must programmatically set the streaming mode in the constructor of the `CPcmToId3Api` class. The first argument of the constructor is of type `CPcmToId3Properties`, which defines the streaming mode as shown in the example below:

```
PCMID3_STREAMING_MODE_E m_nStreamingMode;
```

Suggestions for Follow-up

Check your setting of streaming mode to make certain that it is set to `PCMID3_STREAMING_MODE_LIVE_STREAMING_CONTENT (0).`

Missing INFO tags

Background

The PCM-to-ID3 SDK uses a callback function to deliver ID3 tags to the calling application. The call to `InputAudioData()` does not have a one-to-one relationship with the SDK response in delivering an ID3 tag. In fact, typically many calls to `InputAudioData()` take place between callback reports of ID3 tags that are ready for delivery.

Occasionally, the PCM-to-ID3 SDK delivers two tags immediately following the application's call to `InputAudioData()`: one DATA tag and one INFO tag. In these cases, the client's software may not correctly handle the receipt of two tags at once and drop the INFO tag.

What makes this problem difficult to detect is that it may not occur with every tagging session. When the incoming audio is aligned such that the INFO and DATA tags are delivered after different audio blocks, the problem does not exist.

Symptoms

If many INFO tags are missing, you may see that these Validator reports are marked "FAIL."

```
ID3 Tag: INFO Tag Count:           Fail <xx/0/0>
ID3 Tag: Sequence Number Test:     Fail <0/0/xx>
```

In addition, the log file probably shows the message in the example below at 5-minute intervals:

```
INFO Tag missing on or before sequence number xx, time xxxxx.
```

Note If the content that you are evaluating is less than 6 minutes long, you may not be notified of missing INFO tags. Ask for content that is at least 10 minutes long.

Suggestions for Follow-up

If all or most INFO tags are missing from a file for longer than 10 minutes, make certain that your software is properly handling the arrival of two ID3 tags immediately following delivery of one buffer of audio.

Important If a fail-over tag-inserter device takes over the delivery of HLS segments (including ID3 tags) from the primary inserter device, INFO tags may be dropped. If this transition occurs just before the primary device is scheduled to deliver an INFO tag, an INFO tag or other pending tag may be dropped. Carefully monitor this scenario to determine if more than an acceptable number of tags are dropped.

Improper Set-up of MPEG Multiplexer

Background

If you do not properly set up the PMT of the transport stream to indicate the presence of the ID3 tag PES in the transport stream, neither Validator nor the mobile meters see any Nielsen ID3 tags. Your application must follow the rules—especially the rules for PMT and PES—in the “Timed Metadata for HTTP Live Streaming, Networking & Internet: Protocol Streams.” See “Related Documents” on page 7.

Symptoms

If the MPEG-2 transport stream is improperly set up, you will probably not see any ID3 tags. If Validator is processing transport-stream input, the tests show “Fail” as in the following example:

```
MPEG-2: PMT Descriptor Test:       Fail <0 /5/0>
MPEG-2: PMT Elem Stream Test:      Fail <0 /5/0>
```

Suggestions for Follow-up

Make certain that you have reviewed and complied with the Apple specifications noted above. If an MPEG-2 analyzer is available, use it to examine the PMT for the proper descriptors and the PES headers to make sure that the proper PID is applied to the metadata stream.

ID3 Tags Out-of-Sync with Corresponding Audio

Background

It is **very important** for the ID3 tags to be in sync with the audio from which they were derived. Otherwise, the viewing is improperly credited. For proper synchronization to occur, the PTS of the metadata PES must be set to approximately the same value as the PTS of the audio PES.

Symptoms

An analysis of log files from the mobile meters may show that the tags are greatly shifted from the viewed content.

When Validator processes transport stream input that has both watermarked AC-3 audio **and** ID3 tags, it reports a synchronization error if the tags are separated by more than 15 seconds from the corresponding audio as shown in this example:

```
MPEG-2: Missing PES PTS Test:      Fail <110/0/4>
```

Suggestions for Follow-up

Ensure the PTS of the metadata PES is set to the same value as the PTS of the corresponding audio PES.

Missing or Invalid INFO Tag Settings

Background

One of the final checks that we do before production is ensure that the INFO-tag fields are set correctly by looking at the settings in the Summary Report. The settings appear before the pass/fail list in the Summary Report as shown in the following example and in Figure 2:

```
INFO TAG:
Vendor ID: 999
Device Type: Transcoder or related device
Device ID: 987654321
SystemVersion: 1234
SDK Version: 2.3
Audio CODEC: TestCodec_48000
Distributor ID: www.example.com
```

Missing or Invalid Breakout Type Setting

Background

The Summary Report lists the breakout flag near the top of the report. Validate that the breakout code is set to 0.

```
VALIDATOR SUMMARY REPORT
File evaluated: ID3_TAG_TEST_080614.mpg
Duration: 180 seconds
Breakout Type: 0
FD SID  5131: count = 144
```

Suggestions for Follow-up

Make certain that the end user of the device is correctly setting the breakout type.

Unexpected SIDs in Tags

Background

“Unexpected SIDs” refers to SIDs that are valid but are not expected to appear. (More specifically, there is nothing wrong with the SID itself.) The ID3 tags should reflect the Nielsen SID assigned to the content that is being viewed. The Summary Report lists the PC and FD SIDs represented in all of the tag EDUs:

```
VALIDATOR SUMMARY REPORT
File evaluated: ID3_TAG_TEST_080614.mpg
Duration: 180 seconds
Breakout Type: 3
FD SID  5131: count = 144
```

Suggestions for Follow-up

If some discrepancy in SIDs exists, check the system configuration to ensure that the correct content is being processed and that the audio input is properly configured.

Checking Tag Synchronization in MPEG-DASH Content

When Validator processes MPEG-2 transport streams and HLS streams, if the ID3 tag falls out-of-sync with the audio stream, Validator indicates that Tests 22 and/or 23 failed. If the input is an MPEG-DASH stream, however, Validator is unable to run tests 22 through 30. For MPEG-DASH content, you can manually check tag synchronization by using the Nielsen NACAT application in combination with the ID3 tag listing. Just follow these steps:

1. Process the MPEG-DASH segments using the -df or -dr option.
2. Open the detailed tag-listing file. Find the last ID3 tag that includes 2 or more unstacked EDUs for the same SID. Identify the name of the segment that holds that tag.
3. Identify the names of segments that fall 10 to 20 seconds before the segment identified in step 2. Merge the segment in step 2 with these additional segments (use copy /b, for example) so that you have a 10-20 second segment of audio.
4. Open the NACAT application and, with processing turned on, play the audio stream created in step 3.
5. In the NACAT history on the main window or in the NACAT log files, note the watermarks that NACAT identified.
6. Compare the SIDs and times in the watermarks in NACAT to the SIDs and times of the EDUs identified in step 2. For watermarks with matching SIDs, the timecodes in NACAT should fall within 15 to 20 seconds of the timecodes in the ID3 tag EDUs.

Glossary

A

audio code

See **watermark**.

audio code level

See **distribution type**.

B

breakout code (breakout flag)

The ID3 tag field that identifies the way that advertisements will be inserted into the HLS stream delivered to the computer or mobile device. Set the breakout code to 0. For more information, see the *PCM-to-ID3 SDK Developer's Guide*.

C

CID (Content ID)

An encrypted representation of a watermark SID/timestamp combination Each ID3 tag includes a PC CID (Program Content CID) and an FD CID (Final Distributor CID). The PC CID represents the PC EDUs included in the current ID3 tag. The FD CID represents the FD EDUs included in the current ID3 tag. The timestamp for both CIDs is rolled back to the most recently "crossed" broadcast day start (3 a.m.), relative to the watermark timestamp of the most recent EDU of the same type in the ID3 tag.

D

decoder

An application that decodes an audio stream and reports on Nielsen audio codes found in the stream.

distribution type (or watermarking level, audio code level)

Code that identifies the purpose of the code or watermark:

- FD (Final Distributor): Final distributor of the video content such as a local broadcast affiliate or a cable network.
- PC (Program Content): Identifier of the content itself (often syndication content), not the distributor of the content.
- CC (Commercial Code): Identifier of a commercial or advertisement.
- UK (Unknown): Identifier that indicates difficulty in decoding the watermark so the level is undefined.

DST/Brkout (DST/Breakout)

For the EDU, specifies whether the watermark timestamp was assigned during standard (STD) time or during daylight saving time (DST).

E

EDU (Elemental Data Unit)

A representation of a Nielsen audio code (watermark) that is stored in a Nielsen ID3 tag. There may be up to 10 EDUs in a single Nielsen ID3 tag.

EDU offset

The EDU offset is the eighth field in each EDU line of the ID3 tag listing in the _tags.txt file that Validator generates. The EDU offset specifies when, in relation to the ID3 tag start time, the source watermark was decoded.

F

FD (Final Distributor)

A television broadcaster or network that aggregates and distributes television content to TV viewers. The FDC typically identifies the last distribution outlet for the content. Examples include cable networks like ESPN and CNN and local broadcast stations like WFLA and WABC. Cable and Satellite Multi-System Operators (MSOs) are not really Final Distributors because they simply receive content from a Final Distributor and distribute it to viewers.

FD offset (seconds since FD day-start)

Identifies the number of seconds that separate the most recent valid FD EDU in the current tag from the FD CID timestamp of the tag. The sum of the FD CID timestamp and the PC offset should be within 15 seconds of the most recent FD EDU in the current ID3 tag. In the _tags.txt file, the FD offset is labeled “Seconds Since FD Day-Start.”

H

HLS (HTTP Live Streaming)

The process of including timed metadata (in ID3 format) in an MPEG-2 transport stream. Refer to “Timed Metadata for HTTP Live Streaming.” See “Related Documents” on page 7.

I

ID3 tag

Data format used to carry metadata in an HLS stream. Within the transport or HLS stream, ID3 tags are encapsulated in metadata PES packets.

ID3 tag time

Two time codes are assigned to each ID3 tag: a start time and a tag-release time. The “first” time identifies the time when the Nielsen PCM-to-ID3 software began to collect EDU information for the current ID3 tag. The “last” time identifies the time when the PCM-to-ID3 SDK finalized and packaged the tag and released it to your application. Both tag times are based on the “current time” that your application passes to the PCM-to-ID3 SDK with each call to `InputAudioData().` For the ID3 tag time to be correct, your application must guarantee that the “current time” advances by 1 second for every second of audio data processed.

M

metadata PES

ID3 tags are carried in a metadata PES (packetized elementary stream). The PID of the metadata stream must be listed in the transport-stream Program Map Table (PMT), along with the proper descriptors, as specified in “Timed

Metadata for HTTP Live Streaming." Each Nielsen ID3 tag spans two transport packets; each Nielsen ID3 tag is encapsulated in a single metadata PES packet. See "Related Documents" on page 7.

MPEG-DASH

"DASH is an adaptive bitrate streaming technology where a multimedia file is partitioned into one or more segments and delivered to a client using HTTP." (Source: "Dynamic Adaptive Streaming over HTTP." Wikipedia. Accessed 1017-04-04. https://en.wikipedia.org/wiki/Dynamic_Adaptive_Streaming_over_HTTP)

MPD

MPEG-DASH Media Presentation Description. "A media presentation description (MPD) describes segment information (timing, URL, media characteristics like video resolution and bit rates), and can be organized in different ways such as SegmentList, SegmentTemplate, SegmentBase and SegmentTimeline, depending on the use case.[9] Segments can contain any media data, however the specification provides specific guidance and formats for use with two types of containers: ISO base media file format (e.g. MP4 file format) or MPEG-2 Transport Stream." (Source: "Dynamic Adaptive Streaming over HTTP." Wikipedia. Accessed 1017-04-04. https://en.wikipedia.org/wiki/Dynamic_Adaptive_Streaming_over_HTTP)

N

Nielsen ID3 tag

A specialized ID3 tag, comprised of a single PRIVATE data frame that is used to transmit Nielsen watermark information to Nielsen crediting systems downstream for measuring audience viewing on browsers and mobile devices.

P

PAT (Program Association Table)

A component of an MPEG-2 transport stream that assigns a Packet ID (PID) to each program in the transport stream. After reading the PAT, a transport-stream decoder can use the program PID to access the program map table (PMT) for that program.

PC code (Program Content code)

Code that identifies the distribution sources for the program content before final distribution. Video content can pass through many distribution sources, as when a national broadcast network distributes content to all of its affiliates, before final distribution. Broadcast networks, syndicators, and television studios may each add a PC code to show their place in the distribution chain.

PC offset (seconds since PC day-start)

The PC offset identifies the number of seconds that separate the most recent valid PC EDU in the current tag from the PC CID timestamp of the tag. The sum of the PC CID timestamp and the PC Offset should be within 15 seconds of the most recent PC EDU in the current ID3 tag. In the _tags.txt file, the FD offset is labeled "Seconds Since PC Day-Start."

PES (Packetized Elementary Stream)

Audio, video, and metadata streams are carried within MPEG-2 transport (and HLS) streams in a sequence of transport packets. The transport packets can be sorted by PID (packet ID) into individual audio, video, and metadata streams. Each of these individual elementary streams is referred to as a PES. Nielsen ID3 tags are carried in a metadata PES.

PID (Packet Identifier)

Each 188-byte transport packet in an MPEG-2 transport stream begins with a Packet Identifier (PID) that associates the packet with a specific elementary stream, with a specific table, or with some other component of the transport stream. MPEG-2 decoders use the PID to reassemble the interleaved transport packets into the component to which they belong.

PMT (Program Map Table)

A component of an MPEG-2 transport stream that assigns a PID to all elementary streams in a specific program in the transport stream. For example, the PMT might list the video elementary stream, the audio elementary stream, and the metadata elementary stream. The PMT would assign a PID to each of these elementary streams.

PTS (Presentation Timestamp)

The Presentation Timestamp is the MPEG-2 feature that synchronizes the component elementary streams of the elementary stream. The transcoder application is required to assign to the metadata PES packet the PTS of the audio packet from which the most recent watermark was decoded. In the ID3 tag listing in the _tags.txt file, the PTS field reports the PTS (converted to seconds) of the PES from which the ID3 tag was extracted.

S

sequence number

Each ID3 tag includes an integer sequence number, which starts at 0 when the PCM-to-ID3 SDK is first activated. The sequence number increases by one each time a new tag is released. Both INFO and DATA tags are assigned sequence numbers from the same continuum of integers.

SID (Source ID)

The SID (an integer) uniquely identifies the distribution source or content provider (the entity to which the viewing will be credited). In the _tags.txt file, the second field of each EDU line identifies the SID of the watermark from which the EDU was derived.

T

tag number

Indicates the ordinal position of the current ID3 tag among all of the tags in the detailed tag listing. It reflects the order in which Validator extracted the ID3 tags from the metadata PES.

tag time

See **ID3 tag time**.

TIC (time in content)

For non-linear (file-based, VOD-type) content, the watermark timestamp field is filled with a value called a TIC (time in content). A TIC is the same size (in bytes) as a clock-based timestamp, and, like a clock-based timestamp, it advances by 1 second for every second of audio content that is processed. It does not directly relate to a clock time. Its conversion to a date/time string, while possible, is not meaningful.

timestamp (watermark timestamp, audio code timestamp)

A component of the audio code that is a numeric representation of the date and time at which the watermark was inserted into the stream. The NAES 2 and NAES 6 timestamps are measured from different epochs. Therefore, the NAES 2 raw timestamp for a specific date and time is much larger than the NAES 6 timestamp for the same date and time. In the _tags.txt file, the fourth field of each EDU line identifies the raw timestamp of the watermark from which the EDU was derived. The next-to-last field of the EDU line provides a human readable date and time interpretation of the raw timestamp.

transport packet

All MPEG-2 transport streams are comprised entirely of 188-byte transport packets that can be assembled, based on their PIDs, into elementary streams or into specialized tables.

transport PID

MPEG-2 uses the PID to link transport packets to the elementary PESs to which the packets belong. The transport PID reported in each ID3 tag listing in Validator _tags.txt output file identifies the metadata PES from which the ID3 tag was extracted.

W

watermark

Also known as “audio code.” A sequence of bits that the Nielsen Watermark software periodically inserts into an audio stream. The components of the watermark (SID, timestamp, and level or distribution type) uniquely identify the portion of audio from which the watermark is extracted and the organization (or content) to which the viewing should be credited.

watermark SID

The SID in the source watermark. Note that the EDU a representation of the watermark SID that is stored in an ID3 tag.