

## Usage Guide for DASH-IF Conformance Software

Nomor Research GmbH  
Munich, Germany  
[info@nomor.de](mailto:info@nomor.de)

## Table of Contents

1	Introduction .....	4
2	Usage Guide on DASH-IF Conformance Software .....	4
2.1	Usage via Web User Interface .....	4
2.1.1	Starting Conformance Testing .....	4
2.1.2	During Conformance Testing .....	8
2.1.3	After Conformance Testing .....	10
2.2	Usage via Command Line.....	18
2.2.1	Forming Conformance Test Command .....	18
2.2.2	Running Conformance Test .....	21
2.2.3	Obtaining Conformance Test Results .....	21
3	Usage Guide on Live Conformance Tool.....	23
3.1.1	Accessing Live Conformance Tool.....	23
3.1.2	Starting Live Conformance Testing.....	24
3.1.3	Observing Conformance Testing Results.....	26

## Table of Figures

Figure 1	User Interface of DASH-IF Conformance Software .....	4
Figure 2:	User interface components at the start of conformance testing. ....	5
Figure 3	MPD URL input method .....	6
Figure 4	MPD input using File Upload option .....	6
Figure 5	Optional extension profile enforcement. ....	7
Figure 6:	Optional MPD-only conformance checkbox ticking.....	7
Figure 7:	Running the test. ....	8
Figure 8:	User interface during conformance testing. ....	9
Figure 9	A look of Feature list .....	10
Figure 10:	User interface after conformance test completed. ....	11
Figure 11:	Highlighting all the sections of results.....	13
Figure 12	MPD validation error indication in the result tree .....	13
Figure 13:	Representation error indication in result tree. ....	14
Figure 14	MPD error report .....	15
Figure 15:	Representation error report. ....	15
Figure 16:	Log file downloading via right click. ....	15
Figure 17:	Estimate bitrate user interface.....	16
Figure 18:	Reset to MPD values option. ....	16
Figure 19:	Manually edit values option. ....	17
Figure 20:	Bandwidth estimation given MBT, using default MPD values. ....	17
Figure 21:	MBT estimation given Bandwidth, using manual editing. ....	17
Figure 22:	Conformance test curl command syntax with separate -d flags in case of MPD URL usage.....	19
Figure 23:	Conformance test curl command syntax with concatenated options in case of MPD URL usage.....	19
Figure 24:	Conformance test curl command syntax with separate -F flags in case of MPD file upload usage. ....	20

Figure 25: Conformance test curl command syntax with concatenated options in case of MPD file upload usage. ....	20
Figure 26: Conformance test wget command syntax with separate -d flags in case of MPD URL usage. ....	21
Figure 27: Command line output. ....	22
Figure 28: Live conformance tool user interface. ....	23
Figure 29: Triggering dynamic validation tool from DASH-IF conformance user interface. ....	24
Figure 30: Dynamic service validator when accessed from DASH-IF conformance user interface. ....	24
Figure 31: Dynamic service validation user interface components at the start of conformance testing. ....	24
Figure 32: MPD providing via MPD URL typing. ....	25
Figure 33: MPD providing via using sample MPDs. ....	25
Figure 34: Optional conformance testing parameters section of the user interface. ....	26
Figure 35 Results of the live conformance ....	27

## Table of Tables

Table 1: Command line components. ....	18
--	----

## 1 Introduction

This document presents the usage guide for the DASH-IF Conformance Software tool and Dynamic Service Validation Tool within DASH-IF Conformance Tool. The screenshots of the User Interface (UI) are added to explain how the testing of the DASH content and visualization of the results can be done.

In Section 2, usage guide for the DASH-IF conformance software is presented whereas in Section 3, the Dynamic Service Validation Tool (live conformance tool) usage guide is provided.

## 2 Usage Guide on DASH-IF Conformance Software

Conformance testing can be performed in two ways, namely using user interface and using command line and the following subsections provide details on the usage of these, respectively.

### 2.1 Usage via Web User Interface

The web-based UI of the DASH-IF Conformance Software tool is shown in Figure 1. This page can be accessed either

1. Publicly from the website, <https://conformance.dashif.org> or
2. Locally from the directory, <Conformance-Frontend/Conformancetest.html>

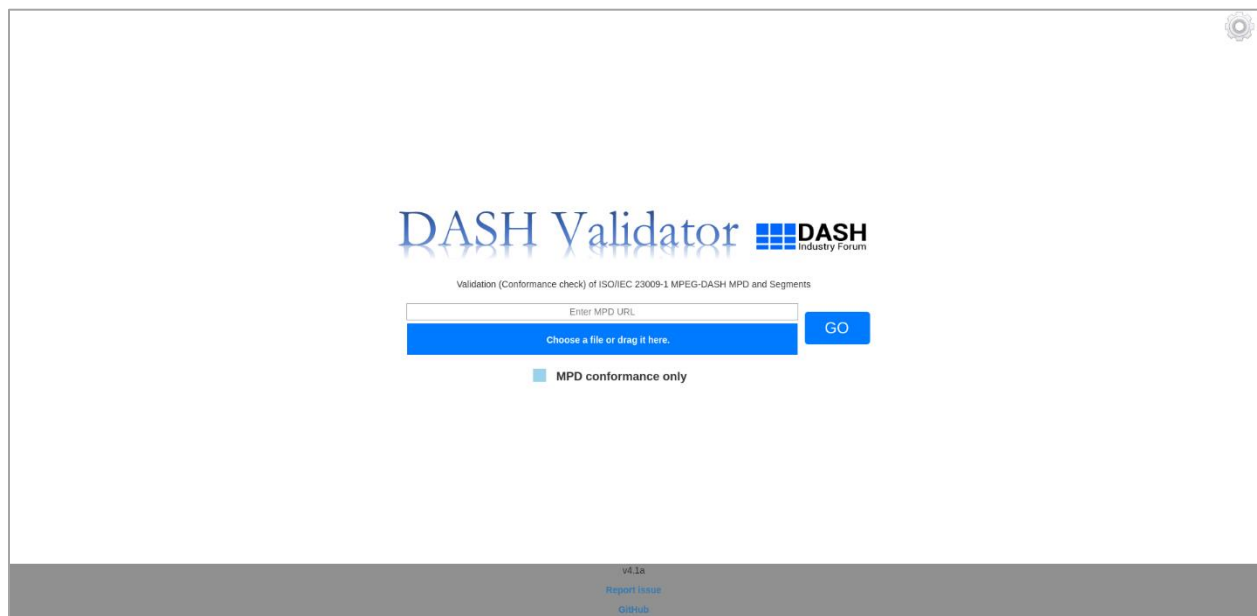
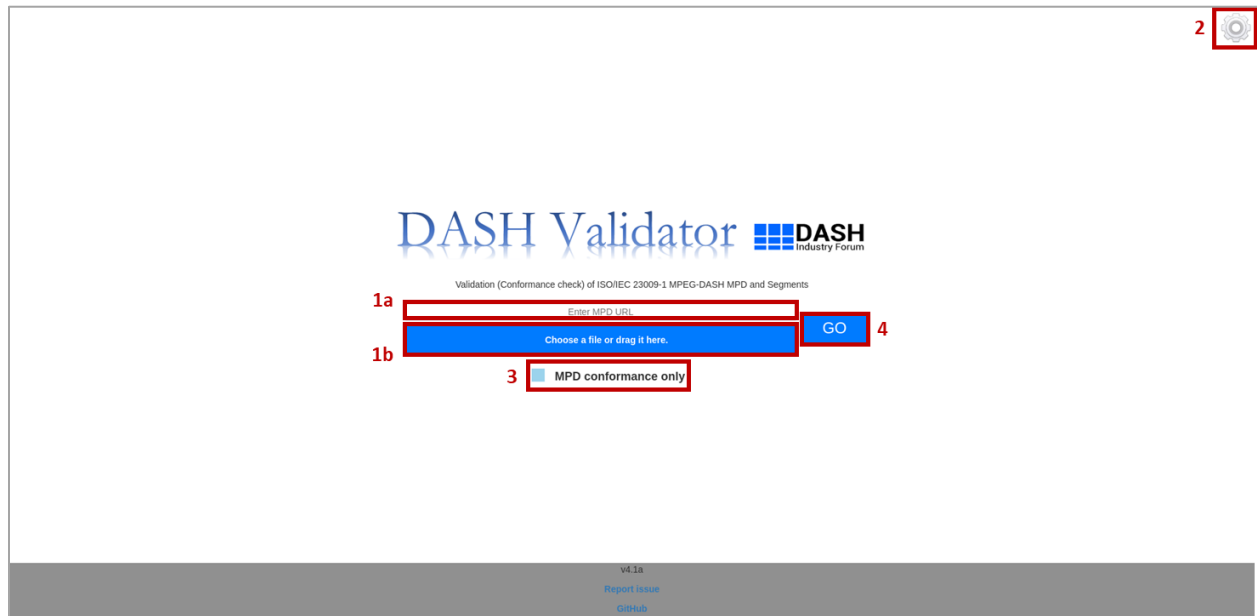


Figure 1 User Interface of DASH-IF Conformance Software

#### 2.1.1 Starting Conformance Testing

At the start of conformance testing, the user interface has the following components as shown in Figure 2.



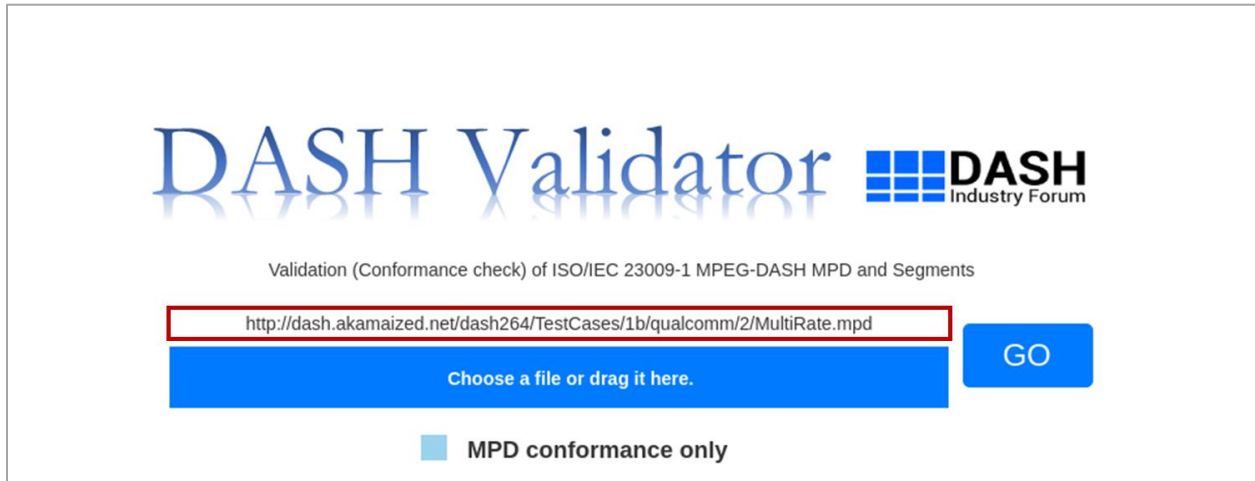
**Figure 2: User interface components at the start of conformance testing.**

These components can be defined as the following:

- MPD URL input bar, denoted by 1a
- MPD file upload bar, denoted by 1b
- Extension profile enforcement option selector, denoted by 2
- MPD-only conformance checkbox, denoted by 3
- Go button, denoted by 4

For starting the conformance testing, we have three steps:

1. **Provide MPD:** This can be done in two ways, specifically, MPD URL providing and local file uploading. MPD URL can be provided in the *MPD URL input bar* as in Figure 3. Alternatively, a local MPD file can be provided to the *MPD file upload bar* by either clicking on 'Choose a file' option or directly dragging the file and dropping it on the highlighted area shown in Figure 4.



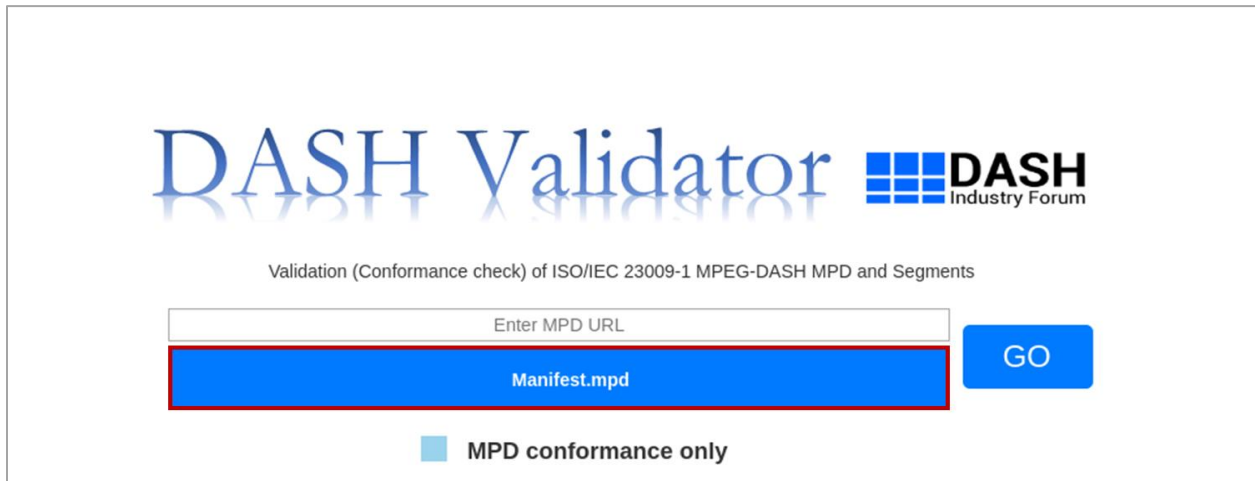
**DASH Validator** 


Validation (Conformance check) of ISO/IEC 23009-1 MPEG-DASH MPD and Segments

Choose a file or drag it here.

☐ MPD conformance only

Figure 3 MPD URL input method



**DASH Validator** 

Validation (Conformance check) of ISO/IEC 23009-1 MPEG-DASH MPD and Segments

Manifest.mpd

☐ MPD conformance only

Figure 4 MPD input using File Upload option

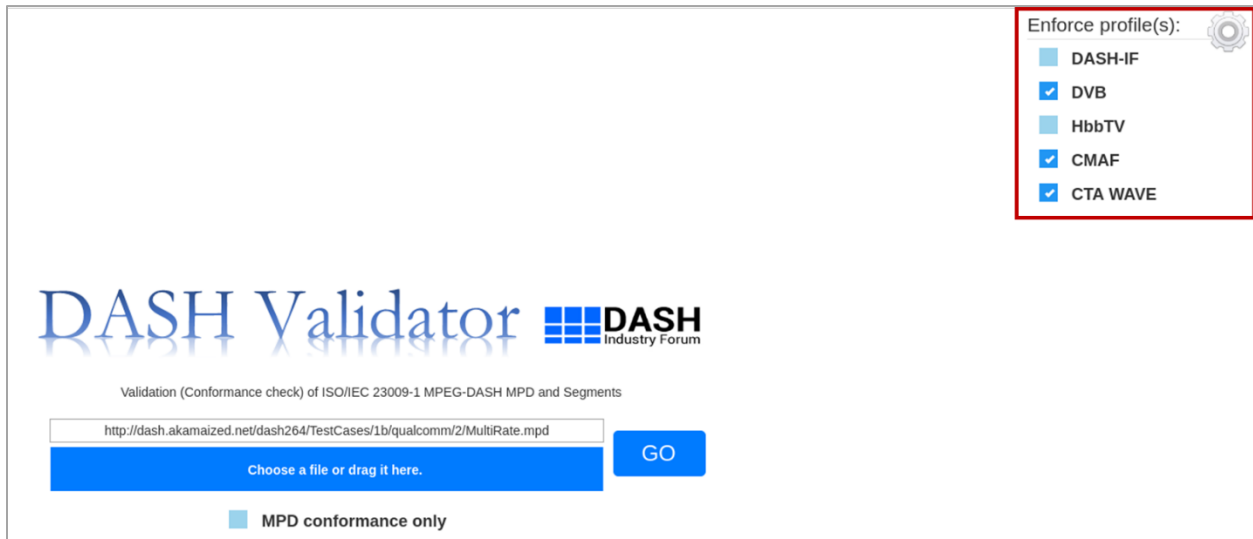
2. **Provide optional conformance settings:** Once the MPD has been provided as described in previous step, one can optionally provide additional conformance settings. These settings correspond to *Extension profile enforcement option selector* and *MPD-only conformance checkbox*. The former – when clicked on – shows additional profiles against which the provided MPD is desired to conformed as shown in Figure 5. Any combination of these profiles can be selected. The latter can be ticked if one desires to only do MPD conformance without conformance testing of media segments pointed to by the MPD. This option is shown in Figure 6.

*Extension profile enforcement option selector* includes the following profiles:

- DASH-IF – <http://dashif.org/guidelines/dash264>
- DVB – <urn:dvb:dash:profile:dvb-dash:2014>
- HbbTV – <urn:hbbtv:dash:profile:isoff-live:2012>

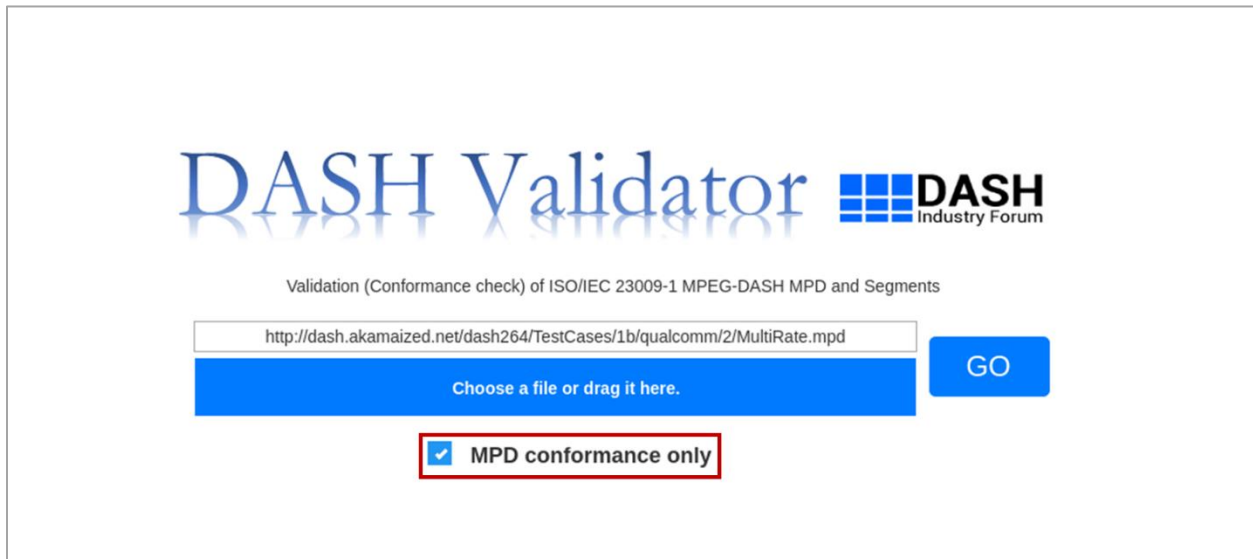
- CMAF – NA
- CTA WAVE – NA

It should be noted that CTA WAVE profile is a superset of CMAF; and therefore, when CTA WAVE profile is enforced, CMAF profile is enforced by default in the backend.



The screenshot shows the DASH Validator interface. At the top right, there is a settings panel titled 'Enforce profile(s):' with a gear icon. It contains five options: DASH-IF (unchecked), DVB (checked), HbbTV (unchecked), CMAF (checked), and CTA WAVE (checked). Below this, the main area displays the 'DASH Validator' logo and the DASH Industry Forum logo. Underneath, it says 'Validation (Conformance check) of ISO/IEC 23009-1 MPEG-DASH MPD and Segments'. There is a text input field containing the URL 'http://dash.akamaized.net/dash264/TestCases/1b/qualcomm/2/MultiRate.mpd'. Below the input field is a blue button labeled 'GO'. At the bottom, there is a checkbox labeled 'MPD conformance only' which is currently unchecked.

Figure 5 Optional extension profile enforcement.



The screenshot shows the DASH Validator interface. It is similar to Figure 5, but the 'MPD conformance only' checkbox at the bottom is now checked and highlighted with a red box. The 'GO' button remains visible.

Figure 6: Optional MPD-only conformance checkbox ticking.

3. **Run the test:** After providing MPD and optionally selecting conformance settings, the last step to start the conformance testing is to click on the *Go button*. When it is clicked on, it will change colour to indicate that the test has started as can be seen in Figure 7.



Figure 7: Running the test.

### 2.1.2 During Conformance Testing

When conformance testing has started, the user interface looks as provided in Figure 8.

The changes that appear on the user interface can be defined as the following:

- Progress status text, denoted by 1
- Profiles information signalled in the MPD, denoted by 2
- Result tree, denoted by 3

During conformance testing, we have three steps:

1. **Observe the conformance test progress status:** The progress information is printed on the user interface to update the user once the conformance test starts. The *progress status text* tells whether the conformance test is in the MPD validation phase or media segment validation phase, and if latter, to what percentage that phase is completed.
2. **Get information on the profiles:** *Profiles information signalled in the MPD* is also printed on the user interface, providing that the conformance testing is being done against these profiles.
3. **Check the on-going results:** The results of the conformance testing get updated continuously after the *Go button* is pressed for each Period in the provided MPD. These are reflected on the *results tree*. When a result section passes the related conformance checks, that part of the section is ticked off with green. Similarly, when there is no error but only warnings related to a conformance check part, that part of the tree is ticked off with green. When there is even a single error, that part of the tree is assigned a wrong sign in red. In the intermediate *result tree* shown in Figure 8, the conformance testing parts are ticked off with green, which means that there were not any errors or warnings.

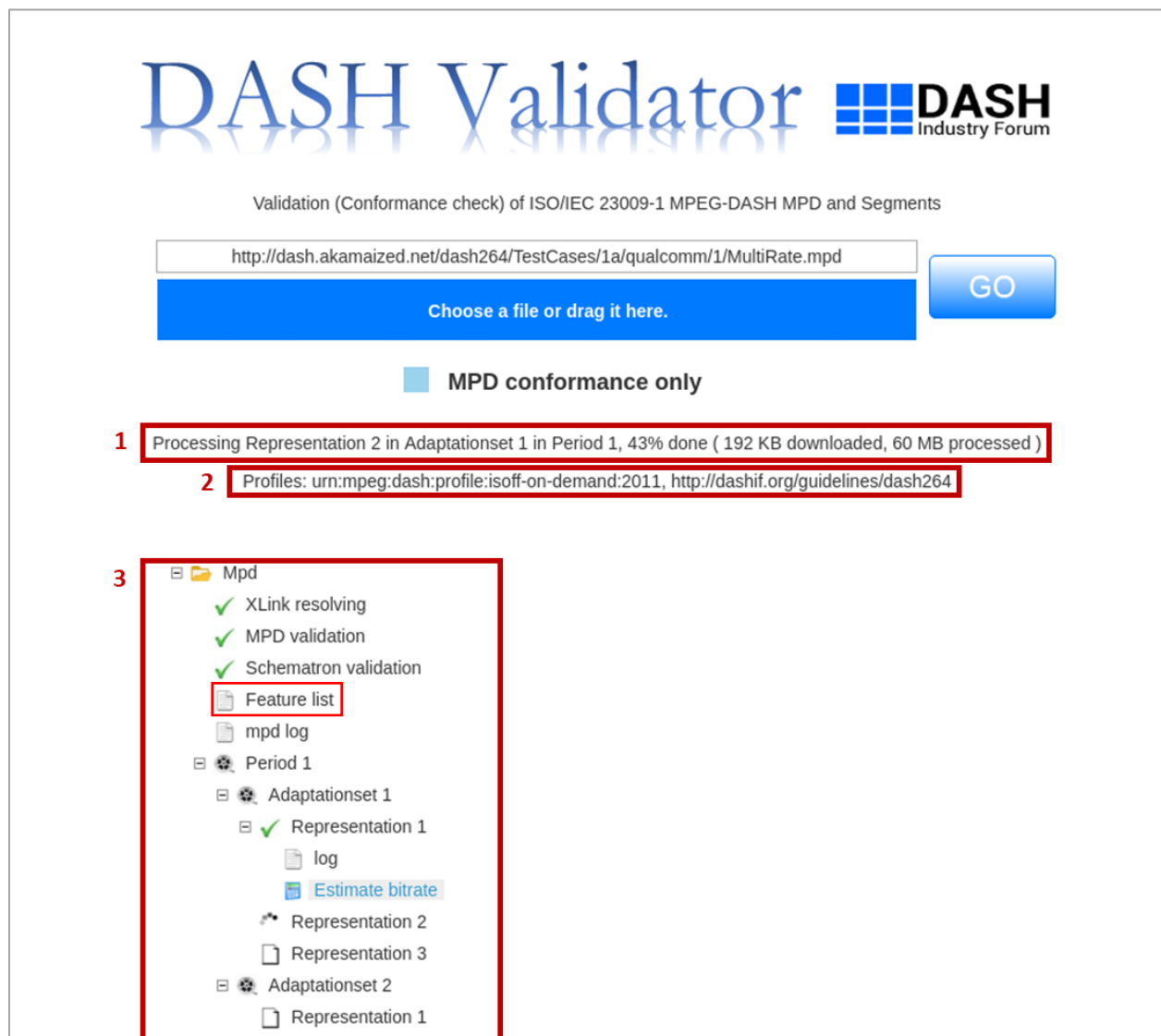
**3.1. Feature list:** To view the list of features, i.e., all the elements and attributes present in the given MPD, click on the link *Feature list* link provided in the *results tree*. When clicked on, a new tab opens with the feature list as depicted in Figure 9.



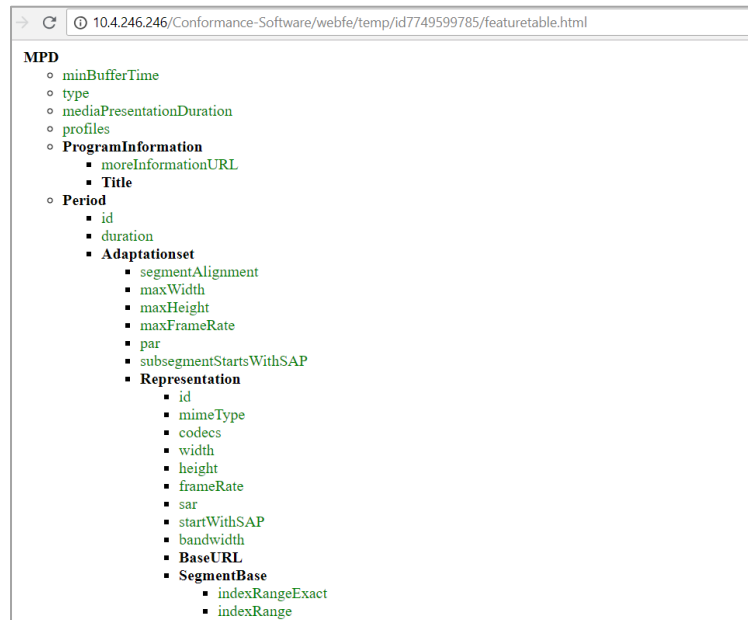
**3.2. Log files:** Each result section is attached a log file which contains the error and/or warning messages concerning the conformance checks or a statement indicating full conformance. The provided logs can be opened by a single click on the log. The respective report opens in a new tab. These files can also be downloaded when right clicked on.

**3.3. Estimate bitrate reporting:** Additionally, each representation part of the tree is attached an “Estimate bitrate” feature, which provides information on possible buffer underrun events when using the specified Minimum Buffer Time (MBT) and Bandwidth. It can be opened by a single click and it opens in a new tab.

For a more detailed explanation of the results please refer to section 2.1.3.



**Figure 8: User interface during conformance testing.**



**Figure 9 A look of Feature list**

### 2.1.3 After Conformance Testing

Once the conformance testing is completed for each Period in the provided MPD, the user interface looks provided in Figure 10.

The final changes appear on the user interface can be defined as the following:

- Conformance test completion status text, denoted by 1
- Conformance result tree final version, denoted by 2

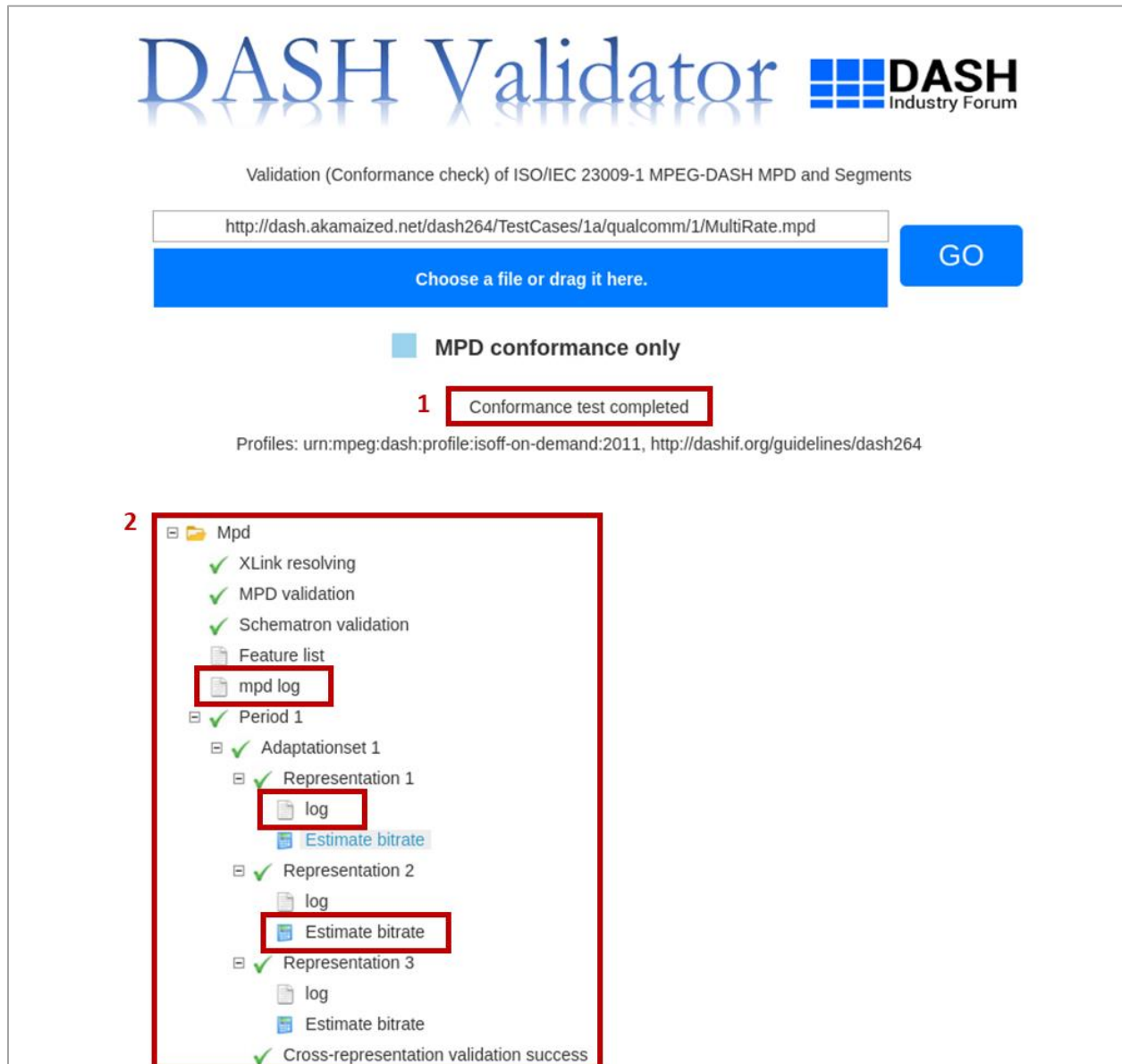


Figure 10: User interface after conformance test completed.

After the conformance testing:

1. **Understand the completion of the test:** Once the conformance testing is completed, it is indicated on the user interface via *conformance test completion status text* stating "Conformance test completed". Hence, this means all the results are available for the user.
2. **Check the results tree:** All the results are available to the user at this stage on the *results tree*. In this step, let us review only the results section of the user interface in detail. The results are formatted in a tree structure. Results can be categorized in three sections as follows:
  - a. **MPD results**

This section consists of “Xlink Validation”, “MPD Validation” and “Schematron Validation” results. When these three parts of this section does not pass the conformance test, even if *MPD-only conformance checkbox* was not selected, the media segment validation is not performed and the conformance test terminates after this section is completed. In case an enforced profile option is selected from *Extension profile enforcement option selector* and that profile contains MPD validation checks, then this section is extended by adding another item on the tree for that specific enforced profile.

**b. Representation/Segments validation results**

This section consists of representation validation results performed for each individual Representation. In the case that *MPD-only conformance checkbox* is ticked, this validation is not performed; and hence, this section does not appear on tree. Given that *MPD-only conformance checkbox* is not ticked and the *MPD results* section passes the conformance checks, then the conformance checks related to this section are performed.

**c. Cross-representation validation results**

This section consists of cross-representation validation results performed for the Representations within the same Adaptation Set. In the case that *MPD-only conformance checkbox* is ticked, this validation is not performed; and hence, this section does not appear on tree. Given that *MPD-only conformance checkbox* is not ticked and the *MPD results* section passes the conformance checks, then the conformance checks related to this section are performed. In case an enforced profile option is selected from *Extension profile enforcement option selector* and that profile contains cross-representation validation checks, then this section is extended by adding another item on the tree for that specific enforced profile.

Figure 11 highlights the abovementioned sections (a, b, c) of results. When a result section passes the related conformance checks, that part of the section is ticked off with green. Similarly, when there is no error but only warnings related to a conformance check part, that part of the tree is ticked off with green. When there is even a single error, that part of the tree is assigned a wrong sign in red. Figure 11 shows an example where all result sections passed all the checks related to them. Figure 12 shows an example of an unsuccessful MPD validation. As can be seen, it is indicated with a wrong sign in front of the respective result section. Figure 13 shows an example of an unsuccessful representation validation.

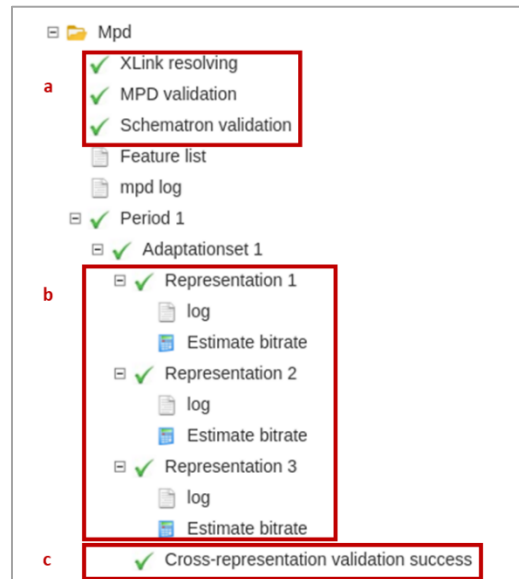


Figure 11: Highlighting all the sections of results.

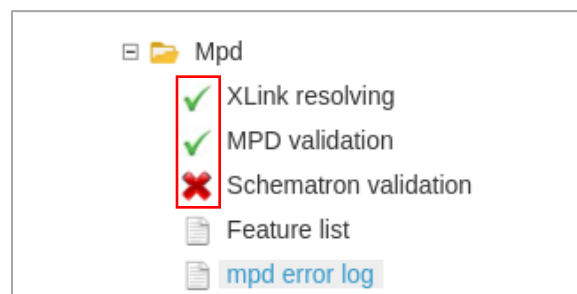
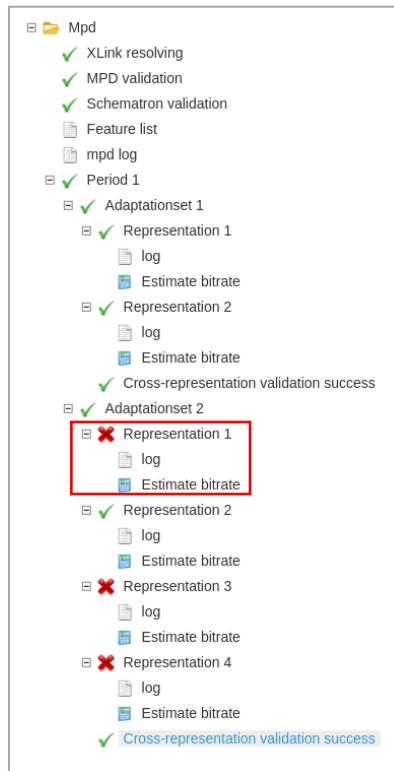


Figure 12 MPD validation error indication in the result tree



**Figure 13: Representation error indication in result tree.**

**2.1. Log files:** Each result section is attached a log file which contains the error and/or warning messages concerning the conformance checks or a statement indicating full conformance. The provided logs can be opened by a single click on the log. The respective report opens in a new tab. An example MPD report shown in Figure 14 highlights the abovementioned error from the Schematron validation part of the section. As can be seen the error provides an error statement and points to the location in MPD where this error occurred. Figure 15 shows a sample representation validation error report. As can be observed, it also consists of detailed information, showing the locations of error (which ISO BMFF box in the representation file) and the error statements.

It should be noticed that the log messages are printed in different colours. As suggested by the “Legend” at the beginning of the reports, blue colour is used for informational messages, yellow is used for indicating a warning and red is used for stating error messages.

```

***Legend: Errors, Warnings, Information ***

Start XLink resolving
=====

XLink resolving successful

Start MPD validation
=====

MPD validation successful - DASH is valid!

Start Schematron validation
=====

location="/*[local-name()='MPD' and namespace-uri()='urn:mpeg:dash:schema:mpd:2011']/*[local-name()='Period' and namespace-uri()='urn:mpeg:dash:schema:mpd:2011']/*[local-name()='AdaptationSet' and namespace-uri()='urn:mpeg:dash:schema:mpd:2011']/*[local-name()='Representation' and namespace-uri()='urn:mpeg:dash:schema:mpd:2011']"
Either the Representation or the containing AdaptationSet shall have the mimeType attribute.
Schematron validation not successful - DASH is not valid!
  
```

Figure 14 MPD error report

```

***Legend: Errors, Warnings, Information ***

ImageDescription name must be '0' not '10'
WARNING: In moov-1:udta-1 - unknown/unexpected atom '©swr'
(27 repetition/s) ### error: moof-1:traf-1:tfhd-1
(27 repetition/s) ### base-data-offset-present is set, violating Section 6.3.4.2. of ISO/IEC 23009-1:2012(E): ... base-data-offset-present shall not be used
error### Buffer underrun conformance error: first (and only one reported here) for sample 1 of run 1 of track fragment 1 of fragment 15 of track id 1 (sample absolute file offset 10183850, fragment absolute file offset 10183850, bandwidth: 2000000)
  
```

Figure 15: Representation error report.

The log files can also be downloaded when right clicked on. In this case, a small context menu appears with only one item, "click to download". For downloading the desired log file in a text file form, please click this item. A screenshot of this feature is provided in Figure 16.

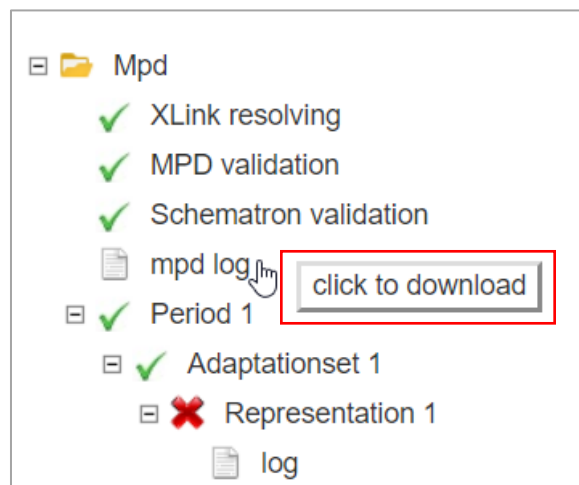


Figure 16: Log file downloading via right click.

**2.2. Estimate Bitrate reporting:** Each representation part of the tree is also attached an “Estimate bitrate” feature, which provides information on possible buffer underrun events when using the specified Minimum Buffer Time (MBT) and Bandwidth. The user interface of this feature is shown in Figure 17.

**Figure 17: Estimate bitrate user interface.**

This page provides the upper or lower boundary within a margin for MBT given the Bandwidth and vice versa such that the buffer underrun does not occur when clicked on ‘Estimate’ button.

- *Radio button:* Either *MinBufferTime* or *Bandwidth* radio button can be selected at a time. The input values by default are set to those found in the MPD for the specific representation.
- *Input bar filling:* From here, one can either *Reset to MPD values* or *Manually edit values*, both of which turn grey when clicked on. Intuitively, *Reset to MPD values* option populates the input bars of both radio buttons with the values from the provided MPD. *Manually edit values* option allows the user to edit any of the input bars. Both these options are shown in Figure 18 and Figure 19, respectively.
- *Estimate:* Two examples depicting the estimating MBT and Bandwidth are provided in Figure 20 and Figure 21, respectively. After the computation is finished this is signalled to the user via turning the input bar of the selected radio button green. Additionally, we provide the margin information on the user interface.

**Figure 18: Reset to MPD values option.**



**Choose a parameter to estimate:**

☐ MinBufferTime (sec) 1.5

☒ Bandwidth (bit/s) 11430

Manually edit values Reset to MPD values Estimate

Figure 19: Manually edit values option.

**Choose a parameter to estimate:**

☐ MinBufferTime (sec) 1.5

☒ Bandwidth (bit/s) 1143014

Manually edit values Reset to MPD values Estimate

Estimation margin 50bps

Figure 20: Bandwidth estimation given MBT, using default MPD values.

**Choose a parameter to estimate:**

☒ MinBufferTime (sec) 1.5366

☐ Bandwidth (bit/s) 1143000

Manually edit values Reset to MPD values Estimate

Estimation margin 100 msec

Figure 21: MBT estimation given Bandwidth, using manual editing.

## 2.2 Usage via Command Line

Conformance testing via command line has following components:

- Software tool for retrieving content with command line utility
- Input
  - Mandatory parameters
  - Optional parameters
- Output

Information on these components are as provided in Table 1. Using these components, the command line conformance testing comprises of 1) conformance test command forming, 2) running conformance test and 3) obtaining the results.

**Table 1: Command line components.**

Components		Information
<b>Software tool</b>		<ul style="list-style-type: none"> <li>• curl</li> <li>• wget</li> </ul>
<b>Input</b>	<b>Mandatory parameters</b>	<ul style="list-style-type: none"> <li>• MPD URL               <ul style="list-style-type: none"> <li>○ url</li> </ul> </li> <li>• MPD local file               <ul style="list-style-type: none"> <li>○ afile</li> </ul> </li> </ul>
	<b>Optional parameters</b>	<ul style="list-style-type: none"> <li>• mpdonly</li> <li>• dashif</li> <li>• cmaf</li> <li>• dvb</li> <li>• hbbtv</li> <li>• ctawave</li> <li>• noerror</li> <li>• nowarning</li> <li>• noinfo</li> <li>• profile</li> </ul>
<b>Execution path</b>		<ul style="list-style-type: none"> <li>• The full path of Process.php file in the local server</li> </ul>
<b>Output</b>		<ul style="list-style-type: none"> <li>• MPD validation result</li> <li>• Representation/segment validation result</li> <li>• Cross-representation validation result</li> </ul>

### 2.2.1 Forming Conformance Test Command

When forming the command one of the two supported software tool should be used and the following subsections provide explanations and examples on each, respectively.

Another important note is to provide at least the mandatory input parameters. The mandatory parameter is regarding the MPD input, which can be provided either in the form of a URL or in the form of a full local path.

Lastly, as the name suggests optional input parameters are optional. Any combination of these parameters can be provided as input in the command. These parameters can be divided into three categories depending on their functionalities:

- **MPD-only conformance testing:** This option can be used if one desires to only do MPD conformance without conformance testing of media segments pointed to by the MPD. The enabler of this functionality is:
  - mpdonly
- **Extension profile enforcement:** This functionality provides additional extension profiles against which the conformance of the provided MPD is desired to be performed. The enablers for this are:
  - dashif – DASH-IF profile extension
  - cmaf – CMAF profile extension
  - dvb – DVB profile extension
  - hbbtv – HbbTV profile extension
  - ctawave – CTAWAVE profile extension
- **Output result suppression:** If one desires to obtain reports with specific type of messages, such as only error messages or no information messages, this feature provides the suppression of specific category of the message types. The enablers of this functionality are as follows:
  - noerror – Suppress error messages in the reports
  - nowarning – Suppress warning messages in the reports
  - noinfo – Suppress informational messages in the reports
- **Media profile validation:** If user wants to validate one or more media profiles from the manifest, *profile* optional parameter can be used. User can specify the media profile using the name ('HD', 'HHD10' etc) or its equivalent 4CC ('cfhd', 'chh1' etc). This feature is currently supported with CTAWAVE profile extension. The usage example is provided in the section 2.2.1.1.1.1.

### 2.2.1.1 Using curl

When using curl, the input parameters are provided with “-d” flag or “-F” flag in case of MPD URL providing and MPD local file uploading, respectively.

#### MPD URL

For this case, “url” parameter is mandatory to provide the MPD file location from which the MPD file will be retrieved. Figure 22 and Figure 23 show the two syntax options for the command forming by providing MPD URL.

```
curl [-d 'url="<MPD_URL>"'] [-d <option1>] [-d <optionN>] <execution_path>
```

Figure 22: Conformance test curl command syntax with separate -d flags in case of MPD URL usage.

```
curl [-d 'url="<MPD_URL>"&<option1>&<option2>'] <execution_path>
```

Figure 23: Conformance test curl command syntax with concatenated options in case of MPD URL usage.

In both cases the <MPD\_URL> is provided between quotation marks. As can be seen the difference between these two syntaxes is the way of providing the input parameters. In the former one each option is provided separately, therefore before each option “-d” flag is present. In the latter one options are concatenated by the use of “&” and therefore only one “-d” flag is present.

Two examples regarding both syntaxes are provided below.

```
curl -d 'url="http://dash.akamaized.net/dash264/TestCases/1a/qualcomm/2/MultiRate.mpd"' -d dashif -d nowarning -d cmf -d mpdonly http://localhost/DASH-IF-Conformance/Utils/Process.php
```

```
curl -d 'url="http://dash.akamaized.net/dash264/TestCases/1a/qualcomm/2/MultiRate.mpd"&dashif&noerror' http://localhost/DASH-IF-Conformance/Utils/Process.php
```

### MPD File Upload

For this case, “afile” parameter is mandatory to provide the MPD file location from which the MPD file will be retrieved. Figure 24 and Figure 25 show the two syntax options for the command forming by providing a local path of the MPD.

```
curl -F 'afile=@<MPD_location>' -F '<option>=1' -F '<option>=1' <execution_path>
```

**Figure 24: Conformance test curl command syntax with separate -F flags in case of MPD file upload usage.**

```
curl -F 'afile=@<MPD_location>' -F '<option>=1&<option>=1' <execution_path>
```

**Figure 25: Conformance test curl command syntax with concatenated options in case of MPD file upload usage.**

As can be seen the difference between these two syntaxes is the way of providing the input parameters. In the former one each option is provided separately, therefore before each option “-F” flag is present. In the latter one, options are concatenated by the use of “&” and therefore only one “-F” flag is present.

Two examples regarding both syntaxes are provided below.

```
curl -F 'afile=@/home/Documents/Manifest.mpd' -F 'mpdonly=1' -F 'dashif=1' -F 'cmf=1' -F 'nowarning=1' http://localhost/DASH-IF-Conformance/Utils/Process.php
```

```
curl -F 'afile=@/home/Documents/Manifest.mpd' -F 'mpdonly=1&dashif=1&cmf=1' http://localhost/DASH-IF-Conformance/Utils/Process.php
```

#### 2.2.1.1.1.1 User specified media profile validation

To validate specific media profile, optional parameter ‘profile’ can be used. For example:

-d 'profile="AAC_Core"'	in case of one media profile specification or
-d 'profile=["AAC_Core", "HD"]'	in case of multiple profiles.

The complete examples using curl are as provided below.

```
curl -d 'url="http://dash.akamaized.net/dash264/TestCases/1a/qualcomm/2/MultiRate.mpd"' -d ctawave -d 'profile="AAC_Core"' http://localhost/DASH-IF-Conformance/Utils/Process.php
```

```
curl -d 'url="http://dash.akamaized.net/dash264/TestCases/1a/qualcomm/2/MultiRate.mpd"' -d ctawave -d 'profile=["AAC_Core","HD"]' http://localhost/DASH-IF-Conformance/Utils/Process.php
```

```
curl -d 'url="http://dash.akamaized.net/dash264/TestCases/1a/qualcomm/2/MultiRate.mpd"' -d ctawave -d 'profile=["aac","cfhd"]' http://localhost/DASH-IF-Conformance/Utils/Process.php
```

The result section contains the informational statements on which track/s conformed with the specified media profile or whether no track conforms.

### 2.2.1.2 Using wget

When using wget, the syntax is as provided in Figure 26. It should be noted that "--post-data" is used to provide all the necessary input. <MPD\_URL> corresponds to the URL at which the MPD file resides. The optional input parameters can be provided in a concatenated form in which case they are separated by "&" symbol.

```
wget --post-data 'url="http://dash.akamaized.net/dash264/TestCases/1a/qualcomm/2/MultiRate.mpd"&<option1>&<option2>' <execution_path>
```

**Figure 26: Conformance test wget command syntax with separate -d flags in case of MPD URL usage.**

One example usage of this is provided below.

```
wget --post-data 'url="http://dash.akamaized.net/dash264/TestCases/1a/qualcomm/2/MultiRate.mpd&mpdonly' http://localhost/DASH-IF-Conformance/Utils/Process.php
```

## 2.2.2 Running Conformance Test

A command line tool of preference should be opened and the directory should be changed to the DASH-IF-Conformance project directory. Here, the formed command should be typed. When pressed "Enter", conformance testing starts.

### 2.2.3 Obtaining Conformance Test Results

After the complete conformance testing finishes for each Period in the provided MPD, the results are printed on the command line as shown in Figure 27.



```

IntegratedConformances curl -n -H "Host: http://10.4.247.94/Server/DASH/264/TestCases/14/qual/connu/77/duf/traf.mpd" -X GET
Error opening file/var/www/html/IntegratedConformance/DASH/counter.txt
***Legend: Errors, Warnings, Information ***
----- MPD Validation Results -----
Start Xlink resolving
-----
Xlink resolving successful

Start MPD validation
-----
MPD validation successful - DASH is valid!

Start Schematron validation
-----
Schematron validation successful - DASH is valid!
-----

----- AdaptationSet 0 Representation 0 Results -----
WARNING: In moov-1:udta-1 - unknown/unexpected atom 'meta'
### error: moov-1
### CMAF check violated (ordinality/nesting) : "In 'moov', the allowed box order as per Section 7.3.1. of ISO/IEC 23000-19(E) is: mvhd--tr
mvhd--mvex--trak--udta
(131 repetition(s) ### error: moof-1:traf-1:tfhd-1
(131 repetition(s) ### CMAF check violated: Section 7.5.14. "Default values or per sample values SHALL be stored in each CMAF chunk's Tra
sample description index' not found.
(31261 repetition(s) ### error: moof-1:traf-1:trun-1
(15696 repetition(s) ### CMAF check violated: Section 7.5.14. "Default values or per sample values SHALL be stored in each CMAF chunk's T
duration' not found in any of them.
(15565 repetition(s) ### CMAF check violated: Section 7.5.14. "default_sample_flags, sample_flags and first_sample_flags SHALL be set in
x to provide sample dependency information within each CMAF chunk and CMAF fragment", not found in any of them.

```

Figure 27: Command line output.

The output results are provided in sections assigned with well-defined names and separated by dashed-lines. The results are printed in different colors. As suggested by the “Legend” at the beginning of the complete report, blue colour is used for informational messages, yellow is used for indicating a warning and red is used for stating error messages. The results comprise of:

- MPD results**  
 This section consists of “Xlink Validation”, “MPD Validation” and “Schematron Validation” results. When these three parts of this section does not pass the conformance test, even if *MPD-only conformance testing* option was not provided, the media segment validation is not performed and the conformance test terminates after this section is completed. In case an enforced profile option is provided from *Extension profile enforcement* and that profile contains MPD validation checks, then this section is extended by adding another item on the tree for that specific enforced profile.
- Representation/Segments validation results**  
 This section consists of representation validation results performed for each individual Representation. In the case that *MPD-only conformance* is provided, this validation is not performed; and hence, this section does not appear on command line. Given that *MPD-only conformance* option is not provided and the *MPD results* section passes the conformance checks, then the conformance checks related to this section are performed.
- Cross-representation results**  
 This section consists of cross-representation validation results performed for the Representations within the same Adaptation Set. In the case that *MPD-only conformance* option is provided, this validation is not performed; and hence, this section does not appear on command line. Given that *MPD-only conformance* option is not provided and the *MPD results* section passes the conformance checks, then the conformance checks related to this section are performed. In case an enforced profile option is selected from

*Extension profile enforcement* and that profile contains cross-representation validation checks, then this section is extended by adding another item on the command line for that specific enforced profile.

### 3 Usage Guide on Live Conformance Tool

In this section, the usage guide for Live Conformance tool or Dynamic Service Validator is presented. The web-based UI of the Dynamic Service Validator tool is shown in Figure 28.



**Figure 28: Live conformance tool user interface.**

#### 3.1.1 Accessing Live Conformance Tool

This page can be accessed either

1. From DASH-IF Conformance User Interface via <https://conformance.dashif.org/Conformance-Frontend/Conformancetest.html> or
2. Publicly from <http://vm1.dashif.org/DynamicServiceValidator>
3. Locally from the directory <DynamicServiceValidator/index.html>

The first access option is triggered when a dynamic-type MPD is detected on the DASH-IF Conformance software. When this happens, the user interface provides a link to the Dynamic Service Validator user interface as shown in Figure 29. When clicked on the link the user interface opens up in a new tab with MPD URL already input to the input bar of the user interface as depicted in Figure 30.



**Figure 29: Triggering dynamic validation tool from DASH-IF conformance user interface.**

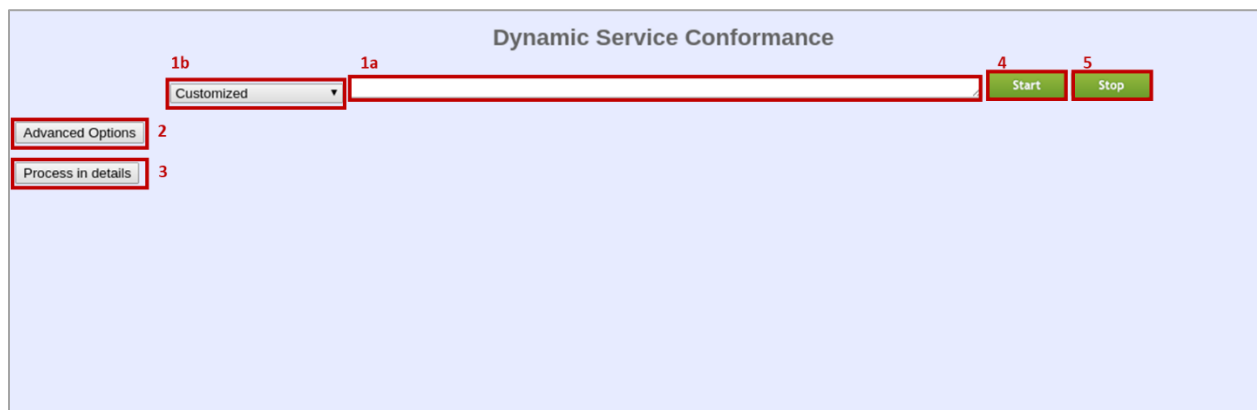


**Figure 30: Dynamic service validator when accessed from DASH-IF conformance user interface.**

The second and third access options are simply opening the user interface via the appropriate URL. In this case, the page does not assign an MPD URL to the URL input bar rather leaves it empty, just as in Figure 28.

### 3.1.2 Starting Live Conformance Testing

At the start of conformance testing, the user interface has the following components as shown in Figure 31.



**Figure 31: Dynamic service validation user interface components at the start of conformance testing.**



These components can be defined as the following:

- MPD URL input bar, denoted by 1a
- Sample MPD dropdown menu, denoted by 1b
- Optional conformance settings, denoted by 2
- Detailed process section, denoted by 3
- Start button, denoted by 4
- Stop button, denoted by 5

For starting the live conformance testing, we have at most three steps:

1. **Provide MPD:** If the access to this user interface is done from the DASH-IF Conformance user interface, this step can be skipped, since the MPD URL will be automatically pasted to the *MPD URL input bar*. If this is not the case, there are two ways of providing an MPD.
  - a. Typing the URL at which MPD resides
  - b. Choosing a sample MPD from the sample MPD dropdown menu

The user interface when these two options are used is shown in Figure 32 and Figure 33, respectively.



The screenshot shows the 'Dynamic Service Conformance' interface. On the left, there are two buttons: 'Advanced Options' and 'Process in details'. The main area features a dropdown menu set to 'Customized' and a text input field containing the URL 'http://vm2.daashif.org/livesim/mup\_300/tsbd\_500/testpic\_2s/Manifest.mpd'. To the right of the input field are 'Start' and 'Stop' buttons.

**Figure 32: MPD providing via MPD URL typing.**



The screenshot shows the 'Dynamic Service Conformance' interface. On the left, there are two buttons: 'Advanced Options' and 'Process in details'. The main area features a dropdown menu set to 'SegmentTimeline' and a text input field containing the URL 'http://livesim.dashif.org/livesim/segmentline\_1/mup\_60/testpic\_2s/Manifest.mpd'. To the right of the input field are 'Start' and 'Stop' buttons.

**Figure 33: MPD providing via using sample MPDs.**

2. **Set optional parameters:** Before running the conformance testing, one can set optional parameters from *optional conformance settings* part of the user interface. When clicked on the user interface looks as shown in Figure 34.



**Figure 34: Optional conformance testing parameters section of the user interface.**

**RTT correction:** For dynamic live services, the segment availability start time (SAST), segment duration (SD) and segment availability end time (SAET) are important concepts for both the segment life time on the server and fetching DASH segments from the server as a client. In the case that the download of the current segment takes long time such that the time advance reaches the SAET or close to SAET of the next segment, the client can still request this next segment as it is in the availability time window. In this regard, the transmission delay of the request may exceed the announced SAET, which results in 404 Not Found message. Round-trip time (RTT) option is provided to avoid this problem. As can be seen from Figure 34, RTT can be manually set to a value in milliseconds or can be decided to be not applied when clicked on the radio button of “None”. By default, RTT of 300 milliseconds is applied.

**Dynamic clock skew correction:** DASH provides many possibilities to assure the synchronization, however there is always a possibility that the synchronization is not exactly trustable caused by problems like real-time process load at the server and server application not running on real-time operating system. In these cases, the SAST timing information would not be accurate, resulting in early requests on the server side. Taking DASH interoperability guidelines into account, a safety margin for such possibility is introduced via this option. When ticked, a clock synchronization margin will be applied for testing SAST and SAET.

3. **Run live conformance testing:** After MPD URL providing and optionally conformance parameters setting, conformance testing can be run by clicking on the *start button*.

### 3.1.3 Observing Conformance Testing Results

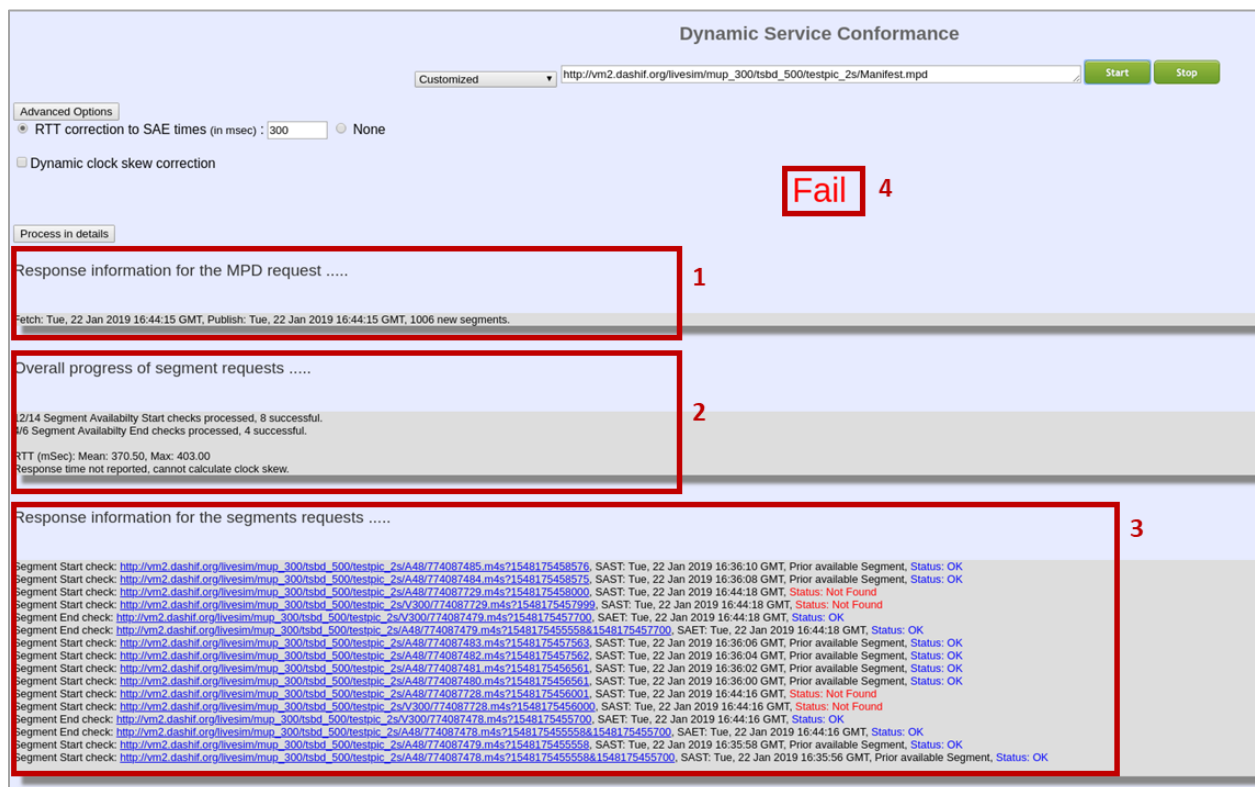
The test results are provided in the *Detailed process section*. When clicked on, the user interface looks like as shown in Figure 35. Results are presented in different sections on the webpage and they are titled as:

1. 'Response information for the MPD request' (displaying MPD fetch and publish times, number of available segments),
2. 'Overall progress of segment requests' (displaying number of successful checks, mean RTT and clock skew),
3. 'Response information for segment requests' (displaying the status of availability start time and end time checks - Status: OK or Not Found).

#### 4. 'FAIL' or 'PASS' (displaying the overall conformance testing result)

"Status: OK" indicates the conformity for the availability start time checks as the segment was available at the availability start time signaled by the MPD. "Status: Not Found" indicates the conformity for the availability end time checks as the segment was not available at the availability end time signaled by the MPD. It should be noted that even a single 404 Not Found message results in 'FAIL' status whereas if there is no 404 Not Found message the test status is displayed as 'PASS'.

Additionally, at any time of conformance testing, one can stop the test by clicking on the *stop button*.



The screenshot displays the 'Dynamic Service Conformance' interface. At the top, there is a dropdown menu set to 'Customized' and a URL field containing 'http://vm2.dashif.org/livesim/mup\_300tsbd\_500testpic\_2s/Manifest.mpd'. To the right of the URL are 'Start' and 'Stop' buttons. Below the URL bar, there are 'Advanced Options' including 'RTT correction to SAE times (in msec): 300' and 'Dynamic clock skew correction'. A large red box labeled 'Fail' with a '4' is positioned in the upper right area of the interface. Below this, there are three sections highlighted with red boxes and numbered 1, 2, and 3. Section 1, titled 'Response information for the MPD request .....', shows 'Fetch: Tue, 22 Jan 2019 16:44:15 GMT, Publish: Tue, 22 Jan 2019 16:44:15 GMT, 1006 new segments.' Section 2, titled 'Overall progress of segment requests .....', shows '2/14 Segment Availability Start checks processed, 8 successful.' and '4/6 Segment Availability End checks processed, 4 successful.' Section 3, titled 'Response information for the segments requests .....', contains a list of segment start and end checks with their respective URLs, timestamps, and status (e.g., 'Status: OK' or 'Status: Not Found').

**Figure 35 Results of the live conformance**