Nomor Research GmbH

Usage Guide for Conformance Software

Nomor Research GmbH
Munich, Germany
info@nomor.de

Nomor Research GmbH
Brecherspitzstraße 8
81541 München

Table of Contents

Table of Figures

Table of Tables

1 Introduction

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

In Section 2, usage guide for the DASH 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 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/index.html

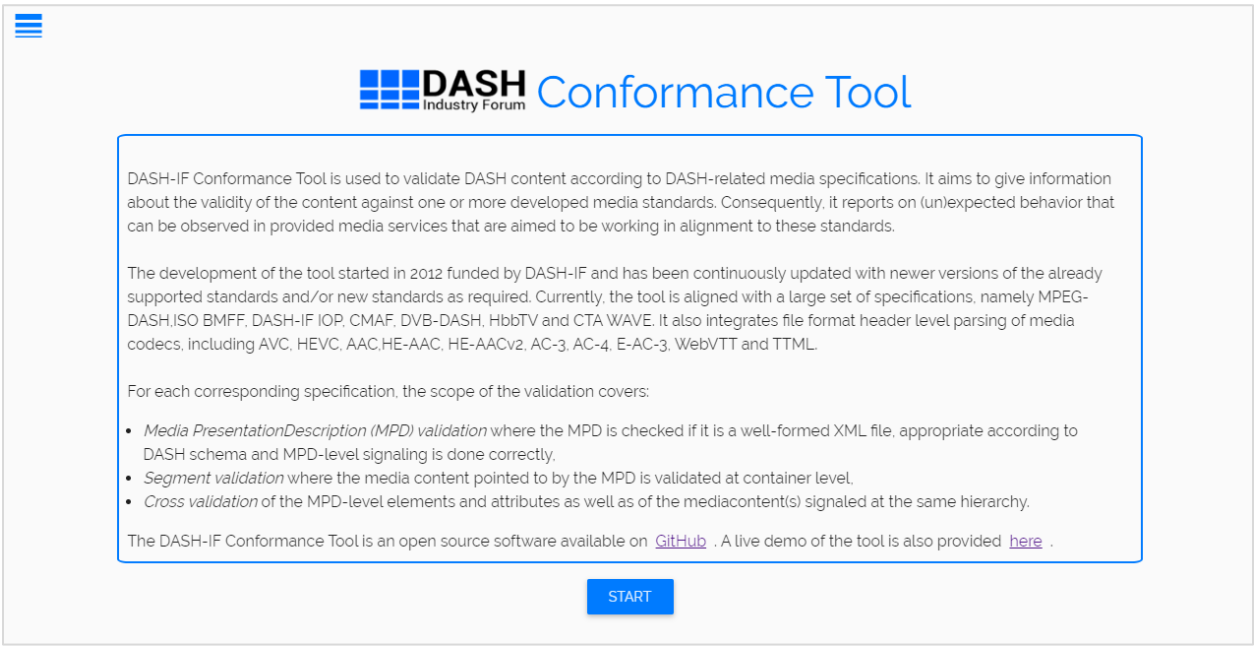

Figure 1: User Interface of DASH-IF Conformance Software

After clicking “START” button on the main page shown in Figure 1, user is redirected to the conformance testing interface. At the start of conformance testing, the user interface has the following components as shown in Figure 2.

These components can be defined as the following:

1. Conformance testing step indicator
2. MPD URL or file upload input bar
3. Additional profile validation options

4. RUN button

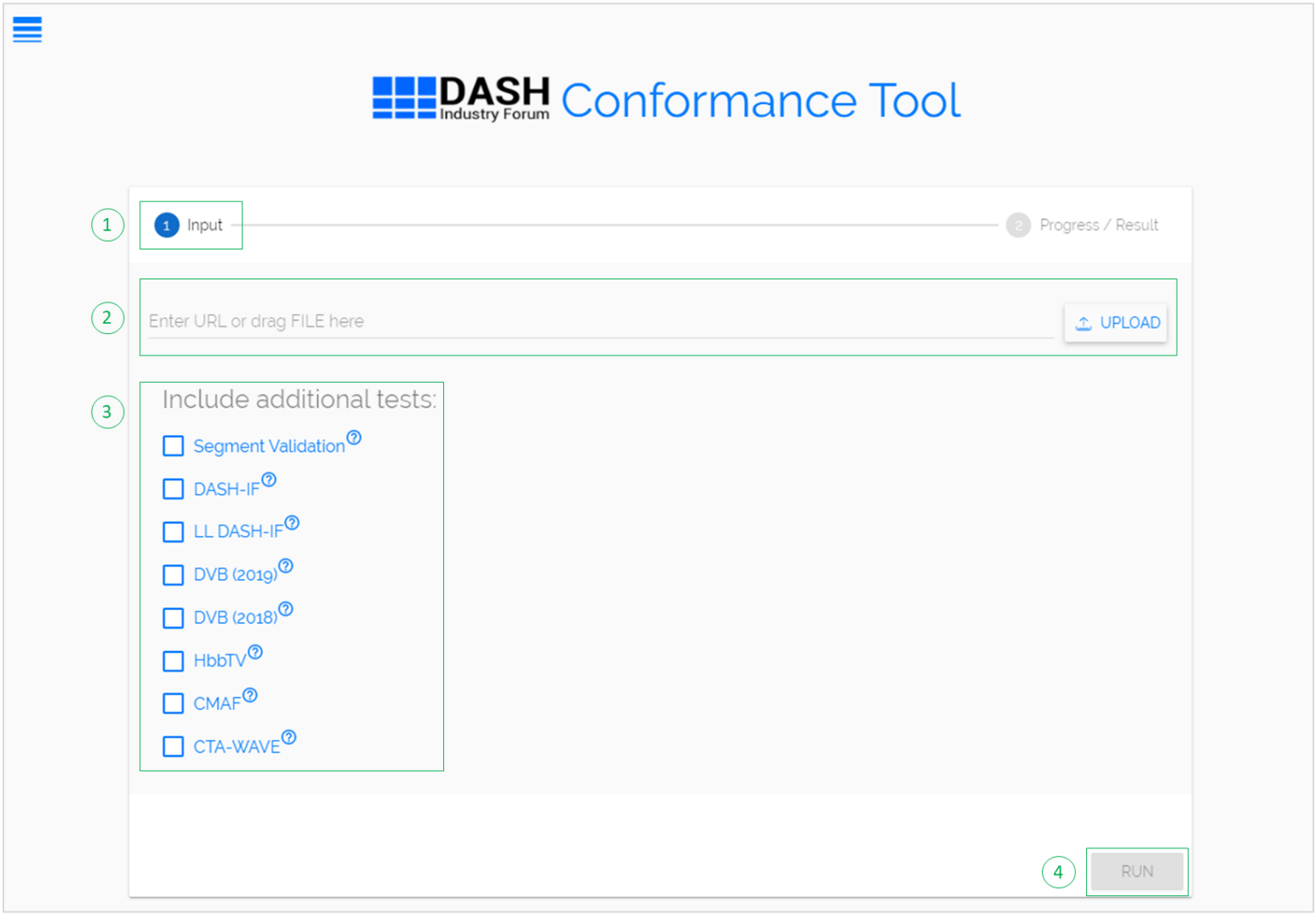

Figure 2: User interface components at the start of conformance testing.

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 directly typed into the input bar and file can uploaded either by clicking on UPLOAD button or by drag&dropping the file in the input bar. Figure 3 shows the MPD URL option.

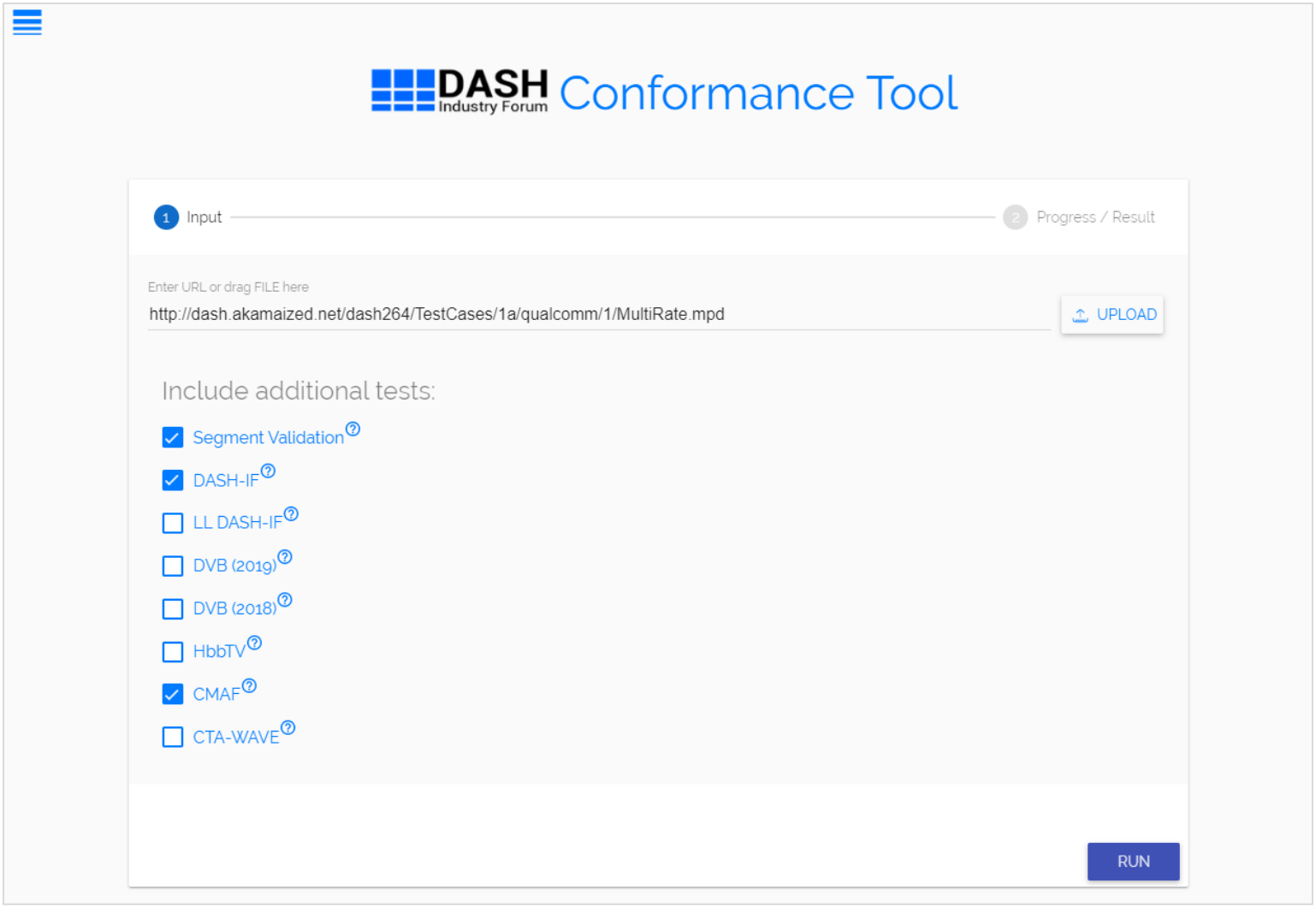

Figure 3: Example conformance testing, input step.

2. ***Provide optional conformance settings:*** Once the MPD has been provided as described in previous step, one can optionally provide additional conformance settings.

   By default, the conformance software provides validation for MPEG-DASH and the profiles included in the MPD@profiles string. The scope of the validation can be extended to include the profiles listed *under "Include additional tests" section in Figure 3.* Any combination of the extensions can be selected and in turn the conformance validation includes the selected extension's results as well in addition to the default conformance operation.

   Additionally, the default validation scope is MPD Validation, which includes DASH MPD schema and MPD-level elements and attributes conformance. Segment Validation option shown in Figure 3 extends the scope to validate the initialization and media segments signalled in the MPD. This includes ISO BMFF and certain supported codecs' validation and covers only the format header level conformance and therefore excludes the elementary stream conformance validation.

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

3. ***Run the test:*** After providing MPD and optionally selecting any number of extension profiles, the last step is to click on the RUN button shown in Figure 2.

When conformance testing starts, the progress and results analysis step is shown to the user as provided in Figure 4.

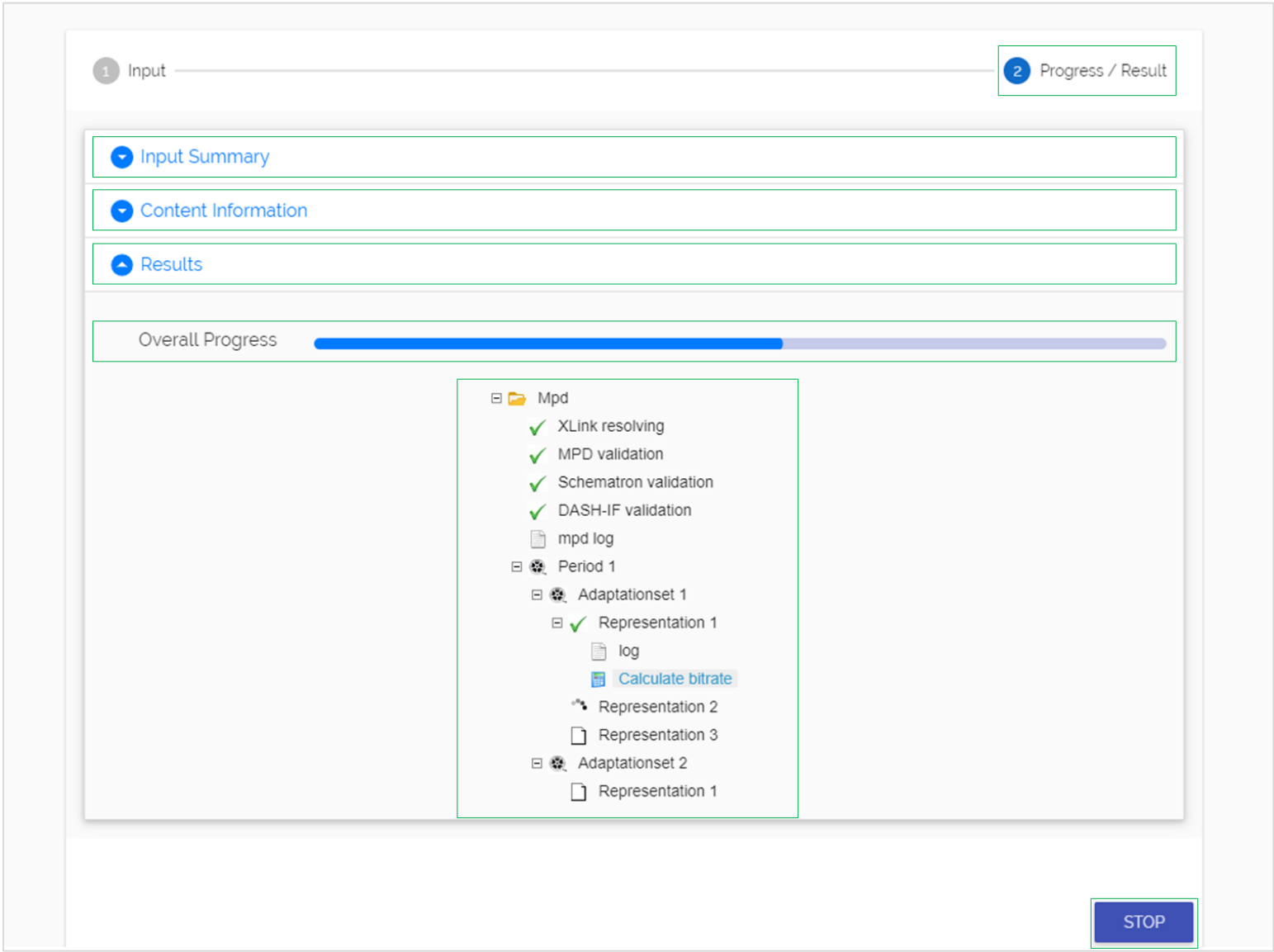

Figure 4: Example conformance testing, progress and results step

The provided information in this step is divided into three main categories:

1. ***Input summary:*** Input information provided on the input step by the user is displayed in this section to provide a summary.

2. ***Content information:*** Profiles information signalled in the MPD is printed in this section to inform the users of the profiles that the MPD will is being validated against. It also provides list of features in MPD, i.e., all the elements and attributes present in the given MPD in a nicely-printed way.

3. ***Results:*** Conformance testing is composed of three steps:
    - *MPD Validation* where the MPD is checked if it is a well-formed file, appropriate according to DASH schema and MPD-level signalling is done correctly,
    - *Segment Validation* where the media content pointed to by the MPD is validated at container level,
    - *Cross Validation* of the MPD-level elements and attributes as well as of the media content(s) signalled at the same hierarchy.

    The results of the conformance testing get updated continuously 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 result tree shown in Figure 4, the conformance testing parts are ticked off with green, which means that there were not any errors or warnings and that the MPD conforms to the profiles.

    It also provides progress information to update the user on which conformance step is being done.

    3.1. 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 as an overlay on the existing page. These files can also be downloaded when right clicked on.
    a. ***MPD Validation 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 Segment Validation option is selected, the conformance test terminates. In case an enforced profile option is selected from extension profiles section and that profile contains MPD validation checks, then the result tree is extended by adding another item on the tree for that specific profile.
    b. ***Segments Validation results:*** This section consists of results for segment and Representation-level validation performed for each individual Representation. In the case that Segment Validation option is not selected, this validation is not performed; and hence, this section does not appear on tree.
    c. ***Cross Validation results:*** This section consists of validation of MPD-level elements and attributes as well as of the media content(s) signaled at the same hierarchy, meaning cross-representation, cross-adaptation set and cross-period validation. In the case that Segment Validation option is not selected, this validation is not performed; and hence, this section does not appear on tree. In case extension profiles are selected and that profile covers cross validation scope, then this section is extended by adding another item on the tree for that specific extension profile.

The results in the log files are color-coded with red indicating conformance error, orange indicating conformance warning and blue indicating conformance information.

3.2. 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.

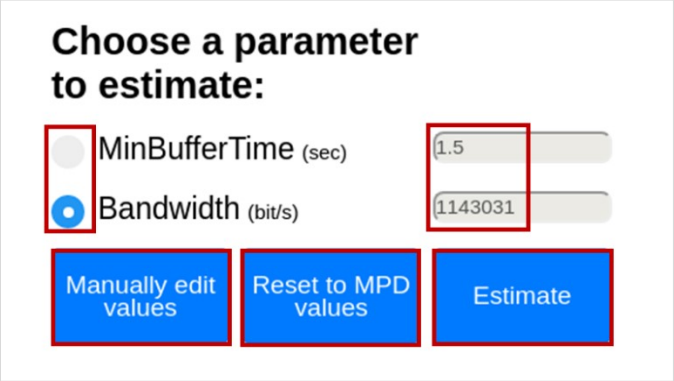

Figure 5: Estimate bitrate user interface.

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*. 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.
- *Estimate:* After the computation is the optimal value with a margin is put onto the computed attributes value bar.

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
Software tool		• curl • wget

Input	**Mandatory parameters**	• MPD URL
		  ○ url
		• MPD local file
		  ○ afile

	Optional parameters	• mpdonly • dashif • cmaf • dvb • hbbtv • ctawave • noerror • nowarning • noinfo • profile
Execution path		• The full path of Process.php file in the local server
Output		• MPD validation result • Representation/segment validation result • Cross-representation validation result

When forming the command one of the two supported tools should be used. 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
  - lldashif – Low Latency DASH 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.

2.2.1 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 6 and Figure 7 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 6: 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 7: 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 cmaf -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 8 and Figure 9 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 8: 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 9: 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 'cmaf=1' -F 'nowarning=1' http://localhost/DASH-IF-Conformance/Utils/Process.php

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

2.2.1.1 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=["caac","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.

2.2.2 Using wget

When using wget, the syntax is as provided in Figure 10. 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="<MPD_URL>"&<option1>&<option2>' <execution_path>
```

Figure 10: 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

For running the validation test, a command line tool of preference should be opened. Here, the formed command should be typed. When pressed "Enter", conformance testing starts. After the complete conformance testing finishes for each Period in the provided MPD, the results are printed on the command line.

The output results are provided in sections assigned with well-defined names. The results in the log files are color-coded with red indicating conformance error, orange indicating conformance warning and blue indicating conformance information.

3 Usage Guide on Dynamic Validation 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 11.

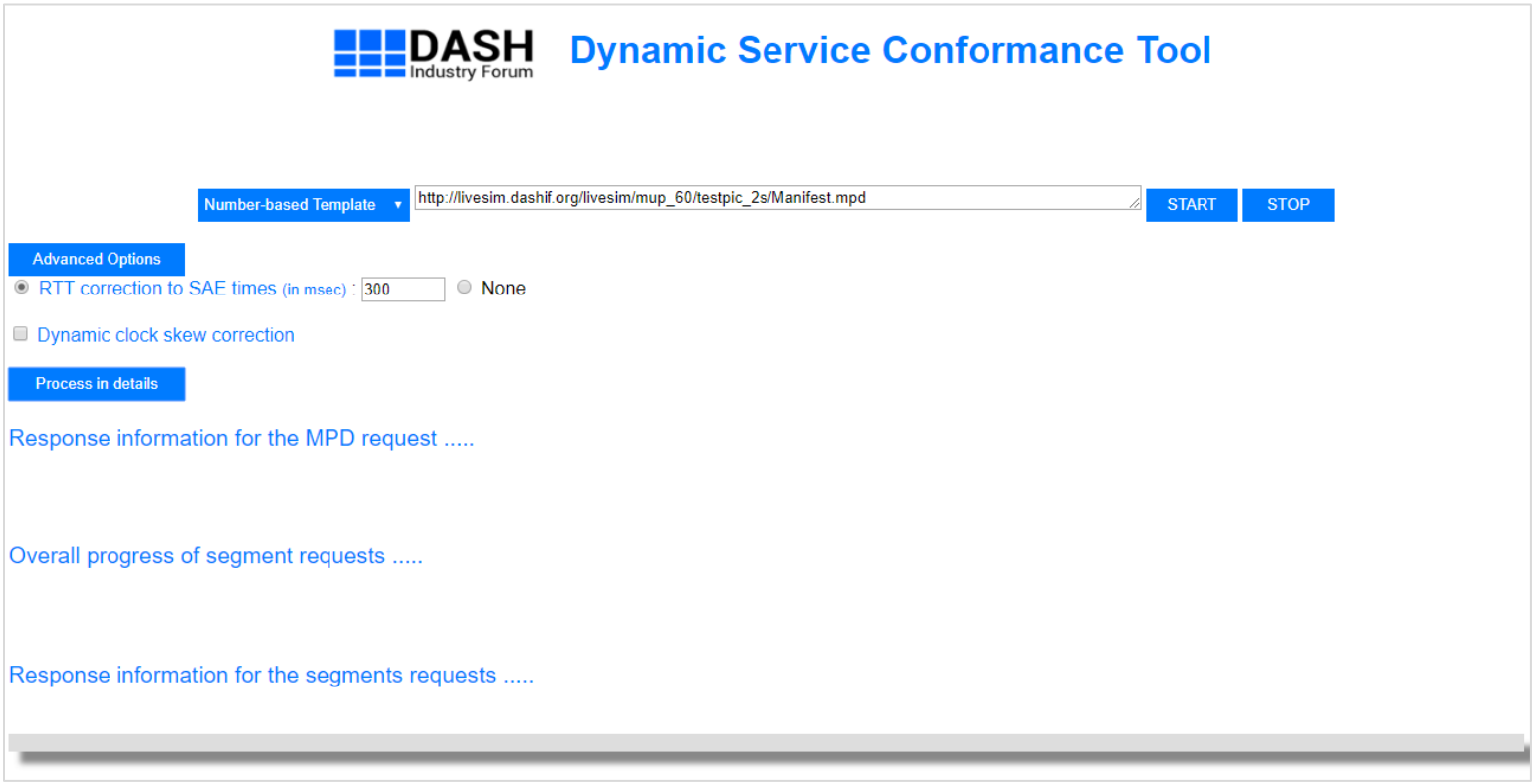

Figure 11: Dynamic Validation tool user interface.

This page can be accessed:

1. DASH-IF Conformance user interface http://conformance.dashif.org/
2. Public page 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 tool. 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 tool. The second and third access options are simply opening the user interface via the appropriate URL.

At the start of conformance testing, the components seen on the interface are as the following:

1. MPD URL input bar
2. Sample MPD dropdown menu
3. Optional conformance settings
4. Detailed process section
5. Start button
6. Stop button

The dynamic conformance testing is composed of 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

2. **Set optional parameters:** Before running the conformance testing, one can set optional parameters from *optional conformance settings* part 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. Its value should be provided in milliseconds and 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*.

The test results are provided in the *Detailed process section*. When clicked on, the user interface looks like as shown in Figure 12. 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).

"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.

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

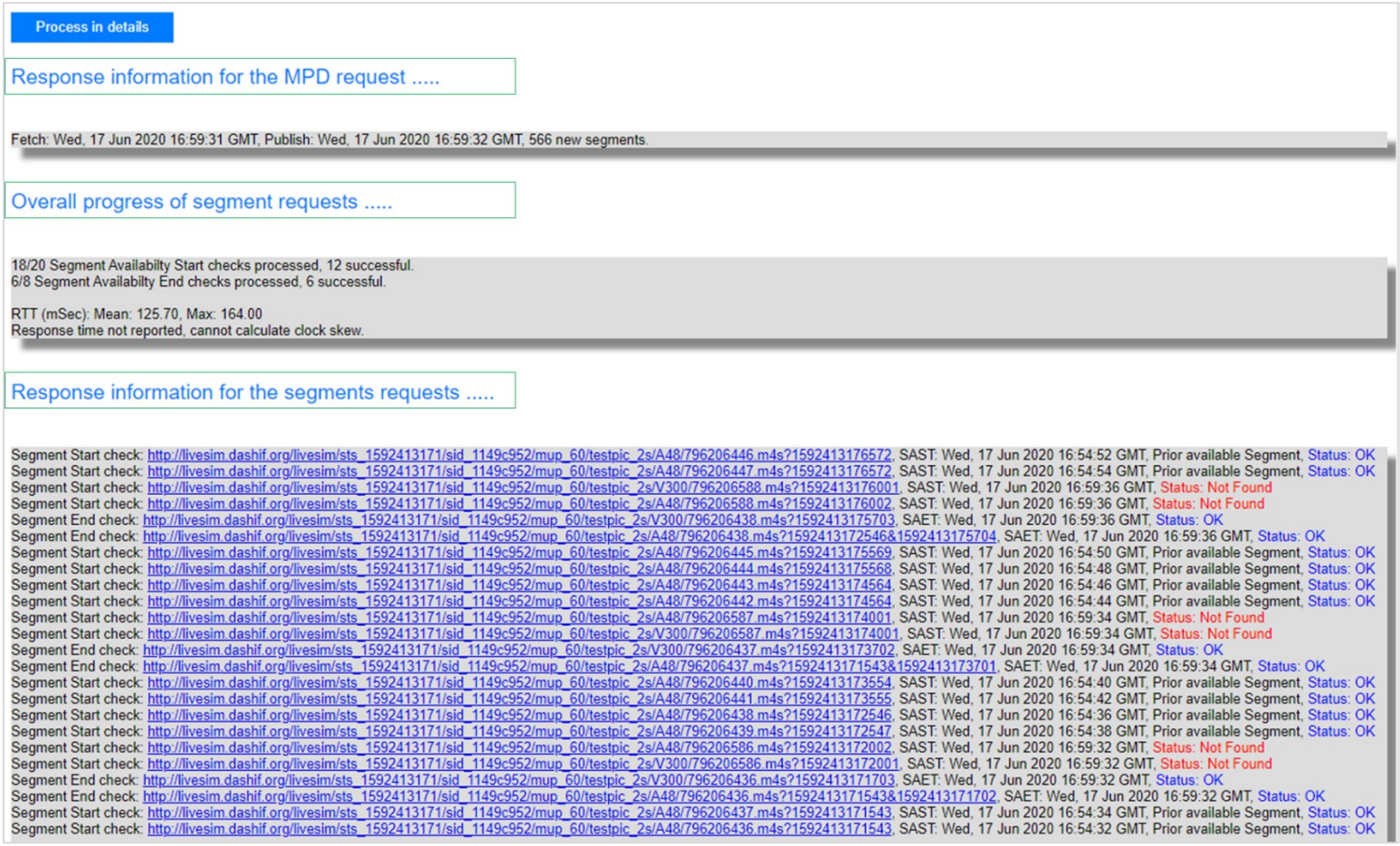

Figure 12 Results of the live conformance