

# 1 Introduction

DASH conformance software performs the task of validating at least the DASH MPD and also the segments pointed by that MPD, as shown in Figure 1 [1]. If there is any conformance issue found, it is reported back. This document provides the software architecture, functionality, and basic usage of the conformance software for DASH and extension to CMAF.

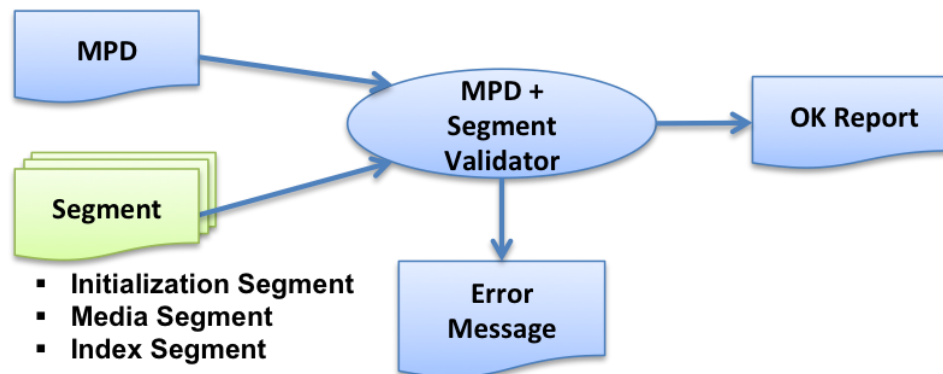


Figure 1: Role of conformance software [1]

Figure 2 shows a relatively detailed process flow of the above-mentioned conformance software. The process flow is described below.

1. Based on the provided location, the MPD is loaded.
2. XML validation and schema validation is done based on the MPD and the MPD schema, the latter pointed to by the MPD.
3. Schematron rules are validated.
4. If segment conformance testing is required, the MPD information is used to access/fetch the segments.
5. ISO BMFF rules are validated.
6. DASH specific rules are validated:
  - a. General DASH rules (e.g. for sidx, ssix boxes etc).
  - b. MPD information specific rules, e.g.
    - i. Profile specific checks,
    - ii. Encrypted content specific checks etc.
7. Cross-representation checks are made (e.g. segment and subsegment alignment).

CMAF conformance can also be verified when the respective mode is selected, which has the following additional process flow as in Figure 2.

1. CMAF specific rules are validated: ISO and non-ISO boxes.
2. Manifest specific checks are validated in coordination with boxes,
  - a. Selection Set, Switching Set, Presentation profile etc.,
  - b. Encrypted content specific checks, Aligned Switching Set checks etc.
3. Cross-CMAF Track checks are validated (e.g. Identical boxes in all Tracks).

Each of the processing green blocks may generate conformance error messages that are provided to the user. Only if none of the blocks generate any error, the content is conforming to the specific DASH/CMAF profile.

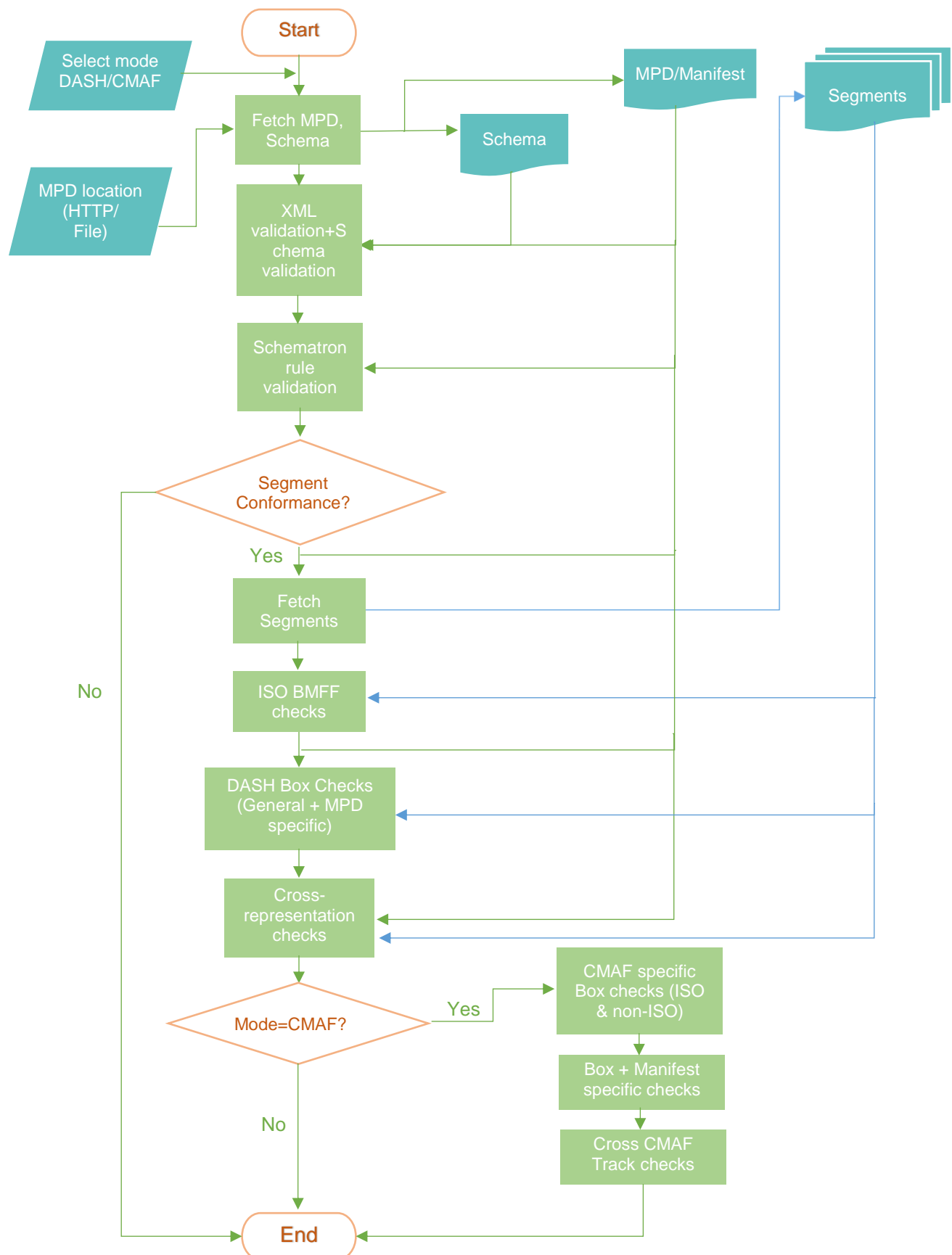
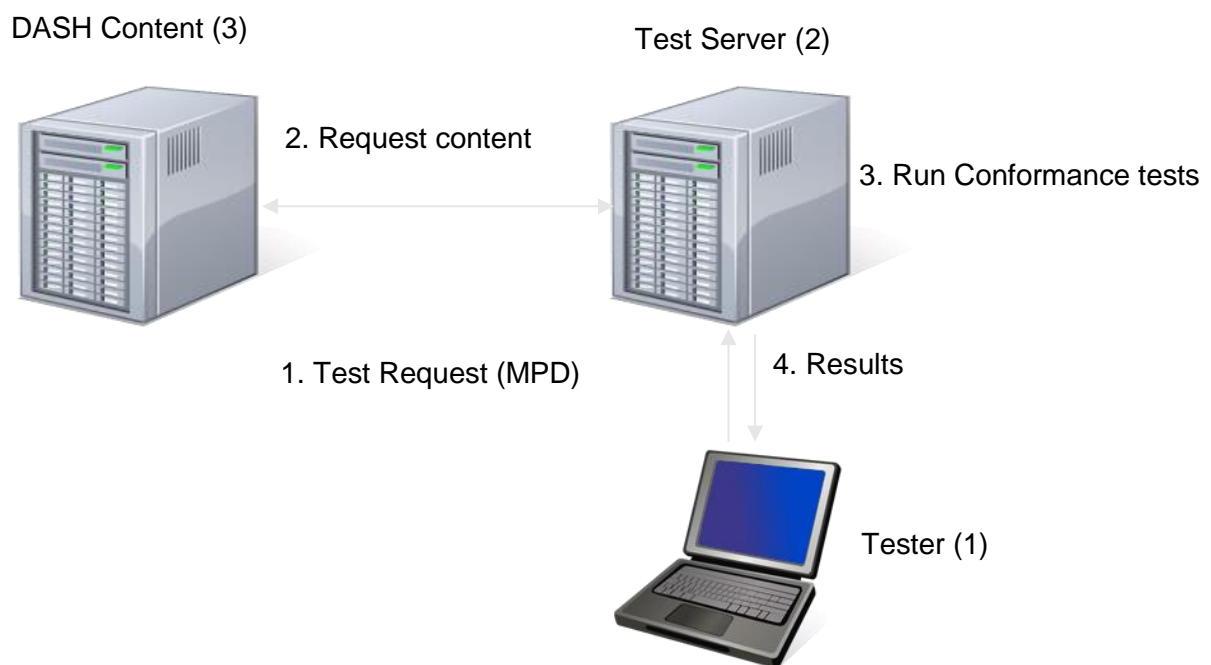


Figure 2: Detailed flow of DASH and CMAF conformance testing

## 2 Realization architecture

Figure 3 shows the functional block diagram of how the DASH conformance software has been realized.

- The conformance software resides on the “Test Server”.
- The “Tester” is a web-client, e.g. Google Chrome browser that access the conformance software on the Test Server.
- The Tested provides the MPD (either as an uploaded file, or points to the MPD location on the web) to the conformance software.
- Based on this information, the conformance software accesses the DASH content.
- Conformance testing is done on the Test Server.
- The report of the conformance testing is provided back to the Tester.



**Figure 3: Functional block diagram of realization architecture**

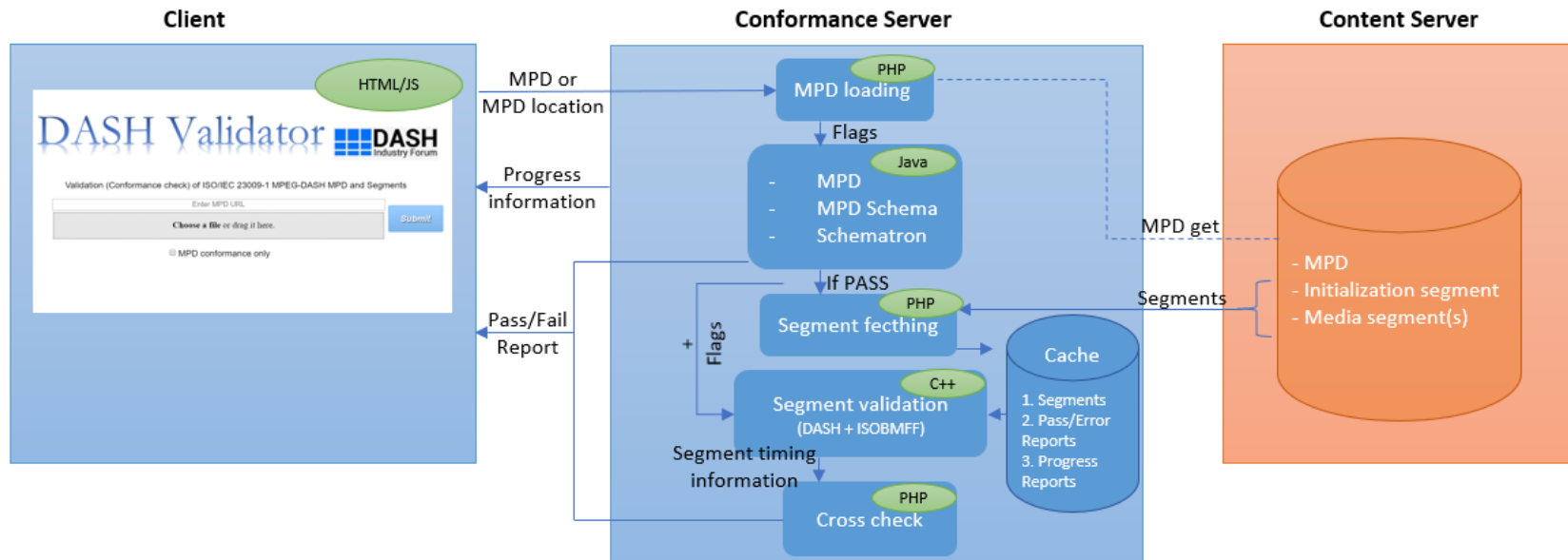
Figure 4 and Figure 5 show detailed functional diagram of MPEG DASH conformance software and its CMAF conformance extension, respectively.

### 2.1.1 Test Server

Linux/Windows based Web server

Frontend code: <https://github.com/Dash-Industry-Forum/Conformance-Software>

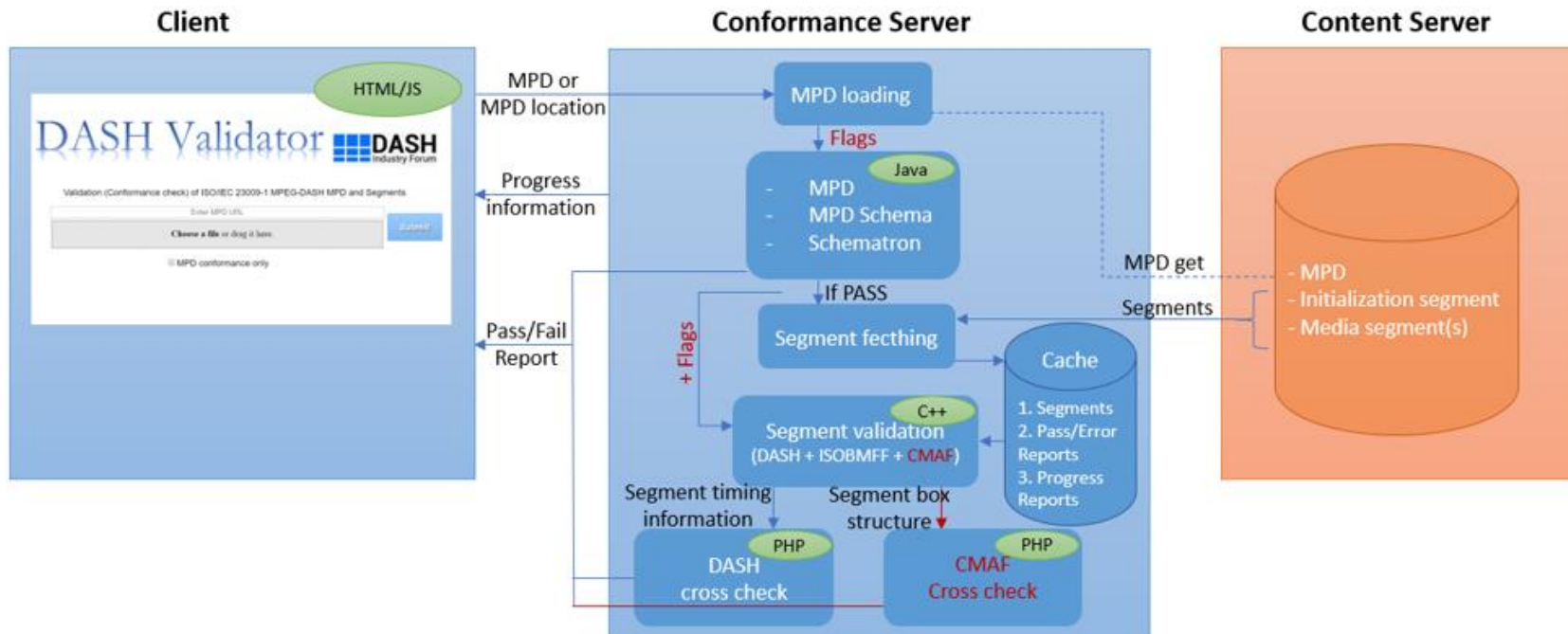
Backend code: <https://github.com/Dash-Industry-Forum/Conformance-and-reference-source>



**Figure 4: Functional diagram of MPEG DASH conformance software tool**

The overall and detailed functional diagram for the conformance software tool is provided in Figure 4.

1. Client accesses the conformance software web interface from a preferred web browser and either uploads the MPD or inputs the MPD URL. Thereafter, the client waits for the information regarding the progress and results.
2. Conformance software loads the uploaded MPD or fetches it from the provided MPD URL.
3. MPD, MPD schema and schematron validation is performed. Error file is generated for client/side reporting.
4. Unless otherwise stated, initialization segment and media segments are fetched from the content server and stored in a local cache.
5. Corresponding MPD information specific flags along with the segments are passed to the segment validation software binary executable, in which ISO-BMFF validation, general MPEG DASH segment rules and MPD information specific segment rules are performed. Error file and information files are generated for client-side reporting.
6. Information files generated by the segment validation software is then used for cross representation validation checks. Error file is generated for client-side reporting.
7. If no errors are found in any of the abovementioned processes, then the provided MPD and the segments are validated to be conformant to the MPEG DASH standard.



**Figure 5: Functional diagram of conformance software tool with CMAF conformance extension**

The overall and more detailed functional diagram of the conformance software tool extended with CMAF conformance (red parts) is as shown in Figure 5.

1. Via the defined interface, CMAF extension takes the local location of the fetched segments and the flags that enable the ISO-BMFF checks as input from the MPEG DASH conformance part.
2. The above input along with additional CMAF flags are passed to the segment validation software binary executable, in which ISO-BMFF validation and CMAF conformance checks are performed. Information file about the segment box structure is generated for each switching set for further cross checks. Error file is generated for client-side reporting.
3. Information files generated by the segment validation software is then used for validation checks about switching sets within selection sets, selection sets and presentations. Error file is generated for client-side reporting.
4. If no errors are found in any of the abovementioned processes, then the provided MPD and the segments are validated to be conformant to the MPEG DASH standards.

### 3 Setup of Test Server

Test server can be installed in Linux/Windows (Linux recommended).

The following packages need to be installed by the specified linux commands.

1. PHP (5.6 or above)  
sudo add-apt-repository ppa:ondrej/php  
sudo apt-get update  
sudo apt-get install php5.6 php5.6-dev php php5.6-curl php5.6-xdebug libapache2-mod-php5.6
2. Install Java

Manually download the newest Oracle (Sun) Java directly from the link

[https://www.java.com/en/download/linux\\_manual.jsp](https://www.java.com/en/download/linux_manual.jsp)

choose the .tar.gz version, unpack them and place at a desired place.

Afterwards setup the java path with the following commands:

```
sudo update-alternatives --install "/usr/bin/java" "java" "path_to_java" 1
```

```
sudo update-alternatives --install "/usr/bin/javaws" "javaws" "path_to_javaws" 1
```

The paths are located in the bin folder of the unpacked package, which is usually called jre1.8.0\_version, so the path\_to\_java could be e.g. /opt/jre1.8.0\_version/bin/java.

Ensure that you also have a java compiler (JRE). This can be verified by using:

```
javac -version
```

Also, ensure that javac and java use the same version. This can be verified by comparing the two commands:

```
java -version
```

```
javac -version
```

If there is no javac present, install the same version as java. For example, the javac 1.8 version could be installed by:

```
sudo apt-add-repository ppa:webupd8team/java
```

```
sudo apt-get update
```

```
sudo apt-get install oracle-java8-installer
```

3. Apache server  
Apache 2.4 is recommended, the root folder is /var/www/html/.  
sudo apt-get install apache2 apache2-doc
4. Configure Apache  
Go to the root folder, copy or move the web contents (DASH/CMAF frontend code) there, or make a softlink of the projects.  
Rename or move the index.html, so that the user defined contents are shown in http://localhost/  
If you want to change the root location, then modify the file /etc/apache2/sites-available/000-default.conf  
When there is no error with the root location, please check if the user or group have corresponding rights. Check the settings by 'ls -l'.  
Make sure that you are in the group www-data and add write permission to the users in this group by:  
sudo chmod -R 0777 /var/www/

(Sometimes addition to the sudoers list is needed. Run `sudo visudo` and add these lines at the end: `www-data ALL=(ALL) NOPASSWD:ALL`)

After any configuration change, a restart of apache service is necessary:

```
sudo service apache2 restart
```

5. Possible additional installations

1. Install 'ant', required to run MPD validator in Conformance-Software.

```
sudo apt-get install ant
```

2. XML extension of PHP

```
sudo apt-get install php5.6-xml
```

3. StdC++ package

```
sudo apt-get install libstdc++6:i386
```

6. If everything works correctly, you should be able to be open the Frontend conformance software in your browser by navigating to:

[http://localhost/CMAF\\_Conformance\\_Frontend/webfe/conformancetest.php](http://localhost/CMAF_Conformance_Frontend/webfe/conformancetest.php)

## 4 Web-user interface

### 4.1.1 4.1 DASH Frontend

The web-based user interface for interaction with the conformance software is shown in Figure 6.

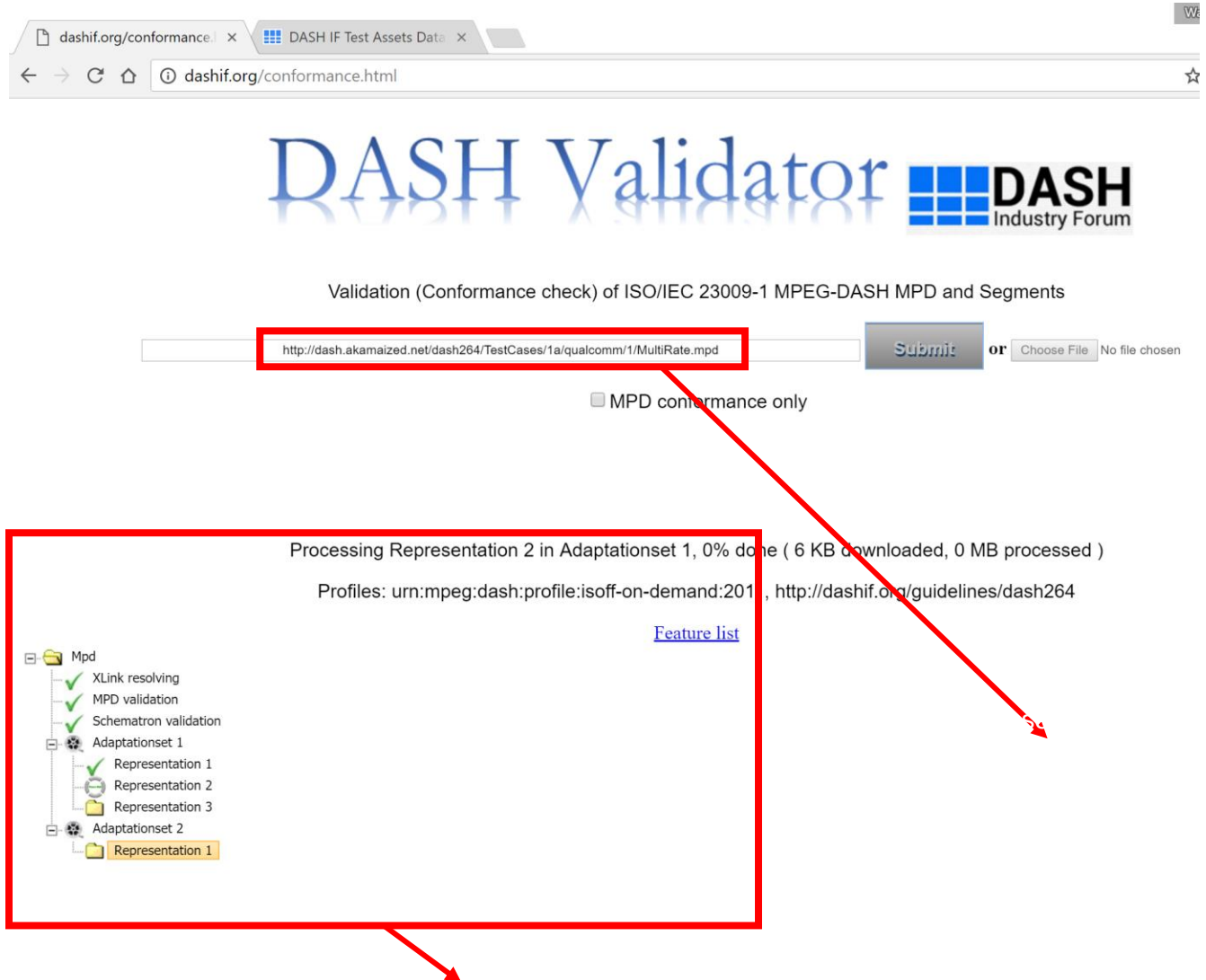


Figure 6: Web user interface of the conformance software.

### 4.1.2 4.2 CMAF Frontend

A query string should be used to initiate CMAF conformance as shown in the Figure 7.



DASH Conformance Test

localhost/CMAF\_Conformance\_Frontend/webfe/conformancetest.php?cmaf=yes

# DASH Validator

DASH Industry Forum

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

<http://dash.akamaized.net/dash264/TestCases/1a/qualcomm/1/MultiRate.mpd> [Submit](#) or [Choose File](#) No file chosen

☐ MPD conformance only

Conformance test completed

Profiles: urn:mpeg:dash:profile:isoff-on-demand:2011, <http://dashif.org/guidelines/dash264>

[Feature list](#)

- Mpd
  - ✓ XLink resolving
  - ✓ MPD validation
  - ✓ Schematron validation
  - ✗ Adaptationset 1
    - ✓ Representation 1
    - ✓ Representation 2
    - ✓ Representation 3
    - ✓ Cross-representation validation success
    - ✗ Compared representations validation error
    - log
  - ✓ Adaptationset 2
    - ✓ Representation 1
    - ✓ Cross-representation validation success
    - ✓ Compared representations validation success
  - ✓ Selection Set
  - ✓ Presentation Profile

CMAF specific results

**Figure 7: Query string usage for enabling CMAF conformance testing**

## References

- [1] ISO/IEC 23009-2, "Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 2: Conformance and reference software"
- [2] Study of DIS of ISO/IEC 23000-19, "Common Media Application Format for Segmented Media", Draft 2017.02.04