

# Content Fulfillment Specification

Version 1.0 31 July 2015

# Content Fulfillment Specification Version 1.0

## Notice:

As of the date of publication, this document is a release candidate specification subject to DECE Member review and final adoption by vote of the Management Committee of DECE in accordance with the DECE LLC Operating Agreement. Unless there is notice to the contrary, this specification will become an adopted "Ecosystem Specification" on 15 September 2015.

THIS DOCUMENT IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE. Digital Entertainment Content Ecosystem (DECE) LLC ("DECE") and its members disclaim all liability, including liability for infringement of any proprietary rights, relating to use of information in this specification. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted herein.

This document is subject to change under applicable license provisions, if any.

Copyright © 2009-2015 by DECE. Third-party brands and names are the property of their respective owners.

## Optional Implementation Agreement:

In addition to the UltraViolet License Agreements which cover implementation of the DECE Ecosystem Specifications within the UltraViolet Ecosystem, DECE offers an optional license agreement relating to the implementation of this document outside the Ecosystem ("RAND Agreement"). Entities executing the optional RAND Agreement receive the benefit of the commitments made by DECE's members to license on reasonable and nondiscriminatory terms their patent claims necessary to the implementation of this document in exchange for a comparable patent licensing commitment. Copies of the license agreements are available at the DECE web site referenced below.

## Contact Information:

Licensing and contract inquiries and requests should be addressed to us at:

<http://www.uvvu.com/uv-for-business>

# Content Fulfillment Specification Version 1.0

## Contents

|       |  |    |
|-------|--|----|
| 1     | Introduction .....                                 | 4  |
| 1.1   | Scope .....  | 4  |
| 1.2   | Document Organization .....                        | 4  |
| 1.3   | Document Notation and Conventions.....             | 4  |
| 1.3.1 | Notations.....                                     | 4  |
| 1.4   | Definitions .....                                  | 5  |
| 1.5   | References .....                                   | 8  |
| 1.5.1 | DECE References .....                              | 8  |
| 1.5.2 | External References .....                          | 8  |
| 1.6   | XML Change Management .....                        | 9  |
| 2     | Content Fulfillment Overview .....                 | 11 |
| 3     | Fulfillment Server .....                           | 12 |
| 3.1   | Network Protocols.....                             | 12 |
| 3.1.1 | HTTP .....   | 12 |
| 3.1.2 | Access Limitations .....                           | 12 |
| 3.2   | Serving Fulfillment Manifest.....                  | 13 |
| 3.3   | Serving Content .....                              | 13 |
| 4     | Fulfillment Client .....                           | 14 |
| 4.1   | Network Protocol .....                             | 14 |
| 4.1.1 | Fulfillment Client Network Requirements .....      | 14 |
| 4.1.2 | Download Client Network Requirements .....         | 14 |
| 4.2   | Obtaining the Fulfillment Manifest .....           | 14 |
| 4.3   | Selecting Components to Download or Stream .....   | 15 |
| 4.4   | CMP Download .....                                 | 15 |
| 4.4.1 | CMP Download and Update Process.....               | 16 |
| 4.4.2 | CMP File Maintenance .....                         | 20 |
| 5     | Fulfillment Manifest File and element .....        | 21 |
| 5.1   | FulfillmentManifest File.....                      | 21 |
| 5.2   | FulfillmentManifest Element .....                  | 21 |
| 5.2.1 | FulfillmentManifest-type .....                     | 21 |
| 5.2.2 | FulfillmentManifestCMP-type.....                   | 21 |
| 5.2.3 | Types used to define CMP Download and Update ..... | 23 |
| 5.2.4 | FulfillmentManifestStreamInfo-type .....           | 26 |
| 6     | Common Download Client and Server .....            | 28 |
| 6.1   | Common Download Server Requirements.....           | 28 |
| 6.2   | Common Download Client Requirements.....           | 28 |

# Content Fulfillment Specification Version 1.0

## 1 Introduction

### 1.1 Scope

This document defines data, called Fulfillment Manifest, and processes for use with the Fulfillment Manifest which supports Common Streaming and DCC (Digital CFF Container) download

### 1.2 Document Organization

This document is organized as follows:

- Section 1      Introduces the organization of this document, and describes its notations and conventions. It includes a glossary of terms, and lists references uses throughout the document.
- Section 2      Defines fulfillment processes
- Section 3      Defines fulfillment data
- Section 4      Defines network protocols used as part of the fulfillment processes
- Section 5      Specifies Common Download Server and Common Download Client

### 1.3 Document Notation and Conventions

#### 1.3.1 Notations

The following terms are used to specify conformance elements of this specification. These are adopted from the ISO/IEC Directives, Part 2, Annex H [ISO-P2H].

SHALL and SHALL NOT indicate requirements strictly to be followed in order to conform to the document and from which no deviation is permitted.

SHOULD and SHOULD NOT indicate that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others, or that a certain course of action is preferred but not necessarily required, or that (in the negative form) a certain possibility or course of action is deprecated but not prohibited.

MAY and NEED NOT indicate a course of action permissible within the limits of the document.

# Content Fulfillment Specification Version 1.0

Terms defined to have a specific meaning within this specification will be capitalized, e.g. “Track”, and should be interpreted with their general meaning if not capitalized. Normative key words are written in all caps, e.g. “SHALL”.

## 1.4 Definitions

|                               |   |
|-------------------------------|---|
| Adaptive Streaming            | Continuous internet download and playback of media Segments while automatically adjusting media bit rate (and possibly encoded picture size) by selecting from alternative Media Segments available from Web servers in order to dynamically adapt bitrate to network throughput without interrupting playback. |
| Asset                         | A component of Content in abstract form (see Logical Asset) or concrete form (see Physical Asset).  |
| Browser                       | Browser is used in these specifications as shorthand for <i>web browser</i> , which is an end-user software application for retrieving, presenting, and traversing information resources on the World Wide Web. A W3C “user agent.”   |
| Common File Format (CFF)      | The standard DECE Content delivery file format, encoded in one of the approved Media Profiles and packaged (encoded and encrypted) as defined by the DECE Common Container & Media Format Specification [DMedia].   |
| Common Streaming              | A framework for delivering stream segments streams conforming with the Common Streaming Format (CSF) definition from a Streaming Server to Streaming Clients where that Streaming Client can play those stream segments.  |
| Common Streaming Format (CSF) | Common Streaming Format (CSF) refers to any Common File Format (CFF) specializations or profiles that are specifically designated for streaming. CSF files CFF files.   |
| Container                     | Shorthand for Digital CFF Container (DCC).  |

## Content Fulfillment Specification Version 1.0

|                              |  |
|------------------------------|--|
| Content                      | A movie, television show, music video, or other media work made available in the Ecosystem. The term Content used informally may include Assets. (In [FRBR], a “work.”)  |
| Content Provider             | Organization providing Content and/or permissions to use Content (e.g., a studio).   |
| DASH                         | Dynamic Adaptive Streaming over HTTP, as specified by [DASH].  |
| DECE                         | Digital Entertainment Content Ecosystem.   |
| Digital CFF Container (DCC)  | An instance of Content published in the Common File Format.  |
| Download Client              | Software that downloads DCCs of CMPs in accordance with this specification. In DECE the Download Client must also comply with [DDevice].   |
| Experience                   | Defines the relationships between Presentations, Applications and other data associated with Content.  |
| Experience Media Application | A Media Application that defines an Experience. An Experience Media Application provides enough information to create a user interface that allows a User to Navigate the Content associated with that Experience. |
| Fulfillment                  | The process of delivering Content to a Fulfillment Client.   |
| Fulfillment Client           | An entity that is capable of providing the client side of downloading or streaming Content in accordance with this specification.  |
| Fulfillment Manifest         | A data structure provided by a Fulfillment Server providing information a Fulfillment Client needs to obtain information necessary to download or stream.  |

## Content Fulfillment Specification Version 1.0

|                                      |   |
|--------------------------------------|---|
| Fulfillment Server                   | An entity that is capable of providing the service side of downloading or streaming Content in accordance with this specification.  |
| Late Binding                         | The combination of separately delivered content (e.g. audio, video, or subtitles) into a single, synchronous presentation.  |
| Logical Asset                        | An abstract instance of Content, independent of the manifestation such as encoding or packaging. (In [FRBR], an “expression.”)  |
| Media Application                    | An application (presentation control program) associated with Content. A Media Application may take forms ranging from simple play lists, declarative data, and markup languages; to procedural language programs that are interpreted by players or virtual machines, or compiled to binary to run on specific processors.       |
| Media Player or Player               | A device or software application that decodes and presents Content from a DCC. A Media.   |
| Media Presentation                   | As defined in [DASH], “collection of data that establishes a bounded or unbounded presentation of media content.” A Media Presentation closely corresponds with a Common File Format DCC, although a Media Presentation can also be an abstract representation of media components that play together through a CMP or Streaming. |
| Media Presentation Description (MPD) | As defined in [DASH], “formalized description for a Media Presentation for the purpose of providing a streaming service.” Note: The formalized description is an XML schema and an instance document of that schema is referred to as an MPD.   |
| Media Profile or Profile             | Requirements and constraints such as resolution and subtitle format for Content in the Common File Format. Current Media Profiles are SD and HD.  |
| Metadata                             | Data that describes Content, including Logical Assets and Physical Assets.  |

# Content Fulfillment Specification Version 1.0

|                     |   |
|---------------------|---|
| Physical Asset      | A specific manifestation of an Asset for a single Media Profile, such as a DCC. (In [FRBR], a “manifestation.”)   |
| Profile             | See Media Profile.  |
| Stream or Streaming | Transmitted Content, protected by an Approved Stream Protection Technology, that is not persistently stored on the receiving LASP Client except for the purposes of buffering, including for instant start of playback, and to enable trick-play. |

## 1.5 References

### 1.5.1 DECE References

The following versions of documents SHALL comprise the version 2.0 DECE “Ecosystem Specifications”:

|              |  |
|--------------|--|
| [DPublisher] | Content Publishing Specification, Version 2.0                                    |
| [DDevice]    | Device Specification, Version 2.0  |
| [DMeta]      | Content Metadata Specification, Version 2.1;<br>mddece-2.0.xsd                   |
| [DCMeta]     | Common Metadata Specification, Version 2.3;<br>md-v2.3.xsd                       |
| [DCMetaCR]   | Common Metadata Ratings Specification, Version 2.2.4                             |
| [DMedia]     | Common File Format & Media Formats Specification, Version 2.0;<br>cff-tt-1.1.zip |
| [DDMP]       | Media Package Specification, Version 2.0   |
| [DStream]    | Common Streaming Protocol Specification, Version 2.0                             |
| [DCManifest] | Common Media Manifest Metadata, Version 1.3;<br>manifest-v1.3.xsd                |

### 1.5.2 External References

|             |  |
|-------------|--|
| [DASH]      | ISO/IEC 23009-1:2014, “Dynamic Adaptive Streaming over HTTP”   |
| [EIDR]      | EIDR ID Format v1.02 <a href="http://eidr.org/documents/EIDR_ID_Format_v1.02_Jan2012.pdf">http://eidr.org/documents/EIDR_ID_Format_v1.02_Jan2012.pdf</a> |
| [HTTP]      | Hypertext Transfer Protocol – HTTP/1.1 (RFC 2616)<br><a href="http://www.ietf.org/rfc/rfc2616.txt">http://www.ietf.org/rfc/rfc2616.txt</a>               |
| [HTTP Auth] | HTTP Authentication (RFC 2617) <a href="http://www.ietf.org/rfc/rfc2617.txt">http://www.ietf.org/rfc/rfc2617.txt</a>                                     |

## Content Fulfillment Specification Version 1.0

|             |   |
|-------------|---|
| [ISO-P2H]   | ISO/IEC Directives, Part 2, Annex H <a href="http://www.iso.org">http://www.iso.org</a>   |
| [RFC3986]   | “Internationalized Domain Names in Applications (IDNA): Protocol”, J. Klensin, August 2010. <a href="http://www.ietf.org/rfc/rfc5891.txt">http://www.ietf.org/rfc/rfc5891.txt</a>   |
| [RFC5891]   | “Uniform Resource Identifier (URI): Generic Syntax” T. Berners-Lee, R. Fielding and L. Masinter, January 2005. <a href="http://www.ietf.org/rfc/rfc3986.txt">http://www.ietf.org/rfc/rfc3986.txt</a>  |
| [SMPTE2053] | SMPTE ST 2053:2011, <i>Media Package for Storage, Distribution and Playback of Multimedia File Sets and Internet Resources</i> , July 13, 2011.   |
| [TLS]       | The Transport Layer Security (TLS) Protocol, Version 1.2 (RFC 5246)<br><a href="http://tools.ietf.org/html/rfc5246">http://tools.ietf.org/html/rfc5246</a>  |
| [URI]       | Uniform Resource Identifier (URI): Generic Syntax (RFC 3986).<br><a href="http://tools.ietf.org/html/rfc3986">http://tools.ietf.org/html/rfc3986</a> and Uniform Resource Identifiers (URIs), URLs, and Uniform Resource Names (URNs): Clarifications and Recommendations (RFC 3305)<br><a href="http://tools.ietf.org/html/rfc3305">http://tools.ietf.org/html/rfc3305</a> |

### 1.6 XML Change Management

XML schemas necessarily change as systems evolve. In particular, Clients and Servers will encounter situations where XML documents are authored against newer revisions of defining XML schemas. It is necessary that Clients and Servers handle this situation gracefully by extracting needed information from the document while ignoring extraneous changes to the document.

In DECE specifications, schema updates are designed to support backwards compatibility. For example, element and attributes can be added, but required elements are not removed; or more generally ordinality of elements and attributes can be widened but not narrowed. Values are not changed in either syntax or semantics.

Given these rules for encoding, servers and clients also follow rules to support backwards compatibility.

An XML document is considered compatible if its structure does not preclude the extraction of data from the document. For example, a document with additional elements and attributes do not preclude schema parsing and data extraction.

For all uses of compatible XML documents, the following requirements apply:

- Client and Servers SHALL NOT reject compatible XML documents, even if they fail schema validation.
- Clients and Servers SHALL extract data from compatible XML documents.

## Content Fulfillment Specification Version 1.0

- Clients and Servers MAY ignore elements and attributes whose presence is not allowed in the specification and schema versions against which that server or client was built. For example, if the original schema allows one instance and three instances are found, the 2<sup>nd</sup> and 3<sup>rd</sup> instance may be ignored.

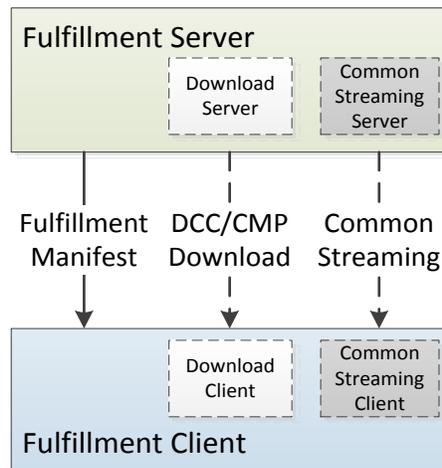
# Content Fulfillment Specification Version 1.0

## 2 Content Fulfillment Overview

This document recognizes several forms of media delivery, primarily DCC (Digital CFF Container) download, CMP (Common Media Package) download and Common Streaming. DCC is defined in [DMedia], CMP is defined in [DDMP] and Common Streaming is defined in [DStream]. There are other forms of delivery such as side-loading (e.g., inserting a flash drive with content) and various proprietary methods. These are not addressed in this specification.

This document refers to the following model. A Fulfillment Server provides Content to a Fulfillment Client. A Fulfillment Server includes a Download Server, defined in this specification, and/or a Common Streaming Server, defined in [DStream]. A Fulfillment Client will have a Download Client, defined in this specification, and/or a Common Streaming Client, defined in [DStream].

The Fulfillment Manifest, defined here, provides data to support the subsequent fulfillment operations, including DCC Download, CMP Download and Common Streaming. Note that additional data is required for Common Streaming as defined in [DStream]. Individual ecosystems using Fulfillment Manifest could have additional constraints on these data and this process.



The Fulfillment process begins with a Fulfillment Client requesting a Fulfillment Manifest from a Fulfillment Server. The Fulfillment Manifest contains information necessary for a Fulfillment Client to initiate Content download or streaming. In the case of download, the Fulfillment Manifest describes all the required and optional components associated with that Content, including references to where that those pieces can be obtained. Generally, these are retrieved from a CDN. In the case of streaming, the Manifest contains a reference to a MPEG DASH [DASH] MPD which will provides streaming information, such as described in Common Streaming Protocol Specification [DStream].

# Content Fulfillment Specification Version 1.0

## 3 Fulfillment Server

The Fulfillment Server is responsible for delivering a Fulfillment Manifest upon request and supplying the download or streaming data associated with that Manifest.

### 3.1 Network Protocols

#### 3.1.1 HTTP

Fulfillment Servers SHALL Support HTTP/1.1 as defined in [HTTP].

Fulfillment Servers SHOULD support TLS as defined in [TLS].

Download Servers SHALL support the HTTP/1.1 GET and RANGE GET commands [HTTP], with or without TLS [TLS], for download of files referenced in the Fulfillment Manifest.

#### 3.1.2 Access Limitations

The Fulfillment Server provides access to Content. In some instances access is not allowed. In these cases an HTTP 403 (forbidden) error response is appropriate. Note that this is independent of any restrictions imposed by content protection such as DRM.

Some examples of access control violations are:

- Payment Required: The Retailer has a policy requiring an additional payment to be made to access content.
- APID Recalled: The Content Provider (licensor) has recalled the APID.
- Fulfillment Restricted: If the Fulfillment Server cannot fulfill a request due to the APID being invalid or due to the ALID being subject to a Fulfillment restriction.
- Access Control: The Retailer does not permit the download to occur. For example, the Content was not purchased.

If the Fulfillment Server cannot fulfill a request due to an access restriction, the Download Server SHALL return an HTTP 403 error response.

# Content Fulfillment Specification Version 1.0

## 3.2 Serving Fulfillment Manifest

A Fulfillment Manifest is obtained by sending an HTTP GET to the correct endpoint (URL) at the Fulfillment Server. The HTTP endpoint of the Fulfillment Server is outside the scope of this document, but must be made known to the Fulfillment Client to begin the Fulfillment process.

Upon request via an HTTP GET, the Fulfillment Server SHALL deliver a Fulfillment Manifest File that is

- Conformant with the definition of Fulfillment Manifest File as defined in Section 5.1
- If a Fulfillment Client processes in accordance with requirements in Section 4, will result in valid Content.
- Contains Location URLs in accordance with Section 3.3.

## 3.3 Serving Content

The Download Server function of the Fulfillment Server downloads CMPs, CMP Parts and potentially other data objects upon request. In this section we refer to these objects as Data Objects.

Each LocationURL in the Fulfillment Manifest provides a location where a Data Object can be downloaded from the Download Server. A Location URL is considered active if its value is a URL referencing an endpoint where the Data Object associated with that Location URL can be downloaded using protocols defined in Fulfillment Server Protocol, Section 3.1. These URLs need to be active long enough to allow download of reference objects (e.g., the OCOMP and all Parts of the CMP), but need not remain active indefinitely. This document does not specify the precise duration.

Multiple instances of LocationURL can be provided within an element associated with a download object (e.g., there can be multiple instances of LocationURL with DCCResource). For this section, these instances are called a LocationURL Set.

The Download Server SHALL ensure at least one LocationURL in Location URL Set is active. That is, at least one LocationURL can be used to download the Data Object.

The Download Server SHOULD ensure that all LocationURL in Location URL Set are active. That is, any LocationURL can be used to download a Data Object.

# Content Fulfillment Specification Version 1.0

## 4 Fulfillment Client

A Fulfillment Manifest contains data that instructs and Fulfillment Client how to download or stream Content.

Some information in the Fulfillment Manifest, such as download links, can be perishable, so it may be necessary to obtain a current Fulfillment Manifest just prior to performing a download or streaming operation.

### 4.1 Network Protocol

This section defines the networking requirements for Fulfillment. Fulfillment requires the use of HTTP. However, it is noted that certain components can be supported via other mechanisms.

#### 4.1.1 Fulfillment Client Network Requirements

Fulfillment Clients SHALL support HTTP/1.1 as defined in [HTTP].

Fulfillment Clients SHOULD support TLS as defined in [TLS].

#### 4.1.2 Download Client Network Requirements

Download Clients MAY use GET or RANGE GET, with or without TLS, to download the files.

Download Clients SHOULD support continuation of downloads that were interrupted.

### 4.2 Obtaining the Fulfillment Manifest

The Fulfillment Client must obtain a URL with the location of the Fulfillment Manifest. How this occurs is outside the scope of this document. It is up to each ecosystem to define its necessary elements of initiation. For example, DECE defines mechanisms for determining the URL through the Coordinator.

The Fulfillment Client SHALL determine the location of the Fulfillment Manifest.

The Fulfillment Client SHALL retrieve the Fulfillment Manifest File, defined in Section 5.1 using an HTTP GET using HTTP as defined in Section 4.1.

If an ecosystem supports other delivery mechanisms for the Fulfillment Manifest, the requirement to use HTTP can be relaxed.

# Content Fulfillment Specification Version 1.0

## 4.3 Selecting Components to Download or Stream

The Fulfillment Manifest can list multiple individual DCCs and CMPs. CMPs can reference optional components. Furthermore, the Fulfillment Manifest can reference optional components to stream.

The Fulfillment Client SHALL determine which optional components to download within the constraints of this specification. How the Fulfillment Client makes this determination is outside the scope of this document.

The Fulfillment Client SHALL determine which optional components to stream provided those options do not conflict with streaming requirements defined in [DStream]. How the Fulfillment Client makes this determination is outside the scope of this document.

## 4.4 CMP Download

The Fulfillment Manifest can include information to support Common Media Package (CMP) downloading. When downloading CMP components, the Download Client first downloads the Original CMP (OCMP). Information in the CMP instructs the Download Client what other assets are required for download, and which are optional. The Download Client downloads the required assets. The Download Client can interact with the User and list the available optional assets for the User to choose from, or select the optional assets automatically based on User preferences (or a combination of both).

The Manifest can include information to support updates. When an APID is replaced (including recalls where the asset is replaced), the replaced APIDs are listed, indicating to the Download Client that an asset has been replaced.

A CMP is updated by revising parts of that CMP while maintaining the same CMP ID. Content Providers publish updates only when the Manifest update mechanism supports those changes; otherwise a new CMP is created, identified by a new CMP ID.

When a CMP has been updated, the Fulfillment Server SHALL include a properly constructed CMP element in the Manifest as follows reflecting the current state of the CMP; and also including indication of obsolete CMP Parts. The specific mechanisms for signaling obsolete Parts is provided later in this section.

The Manifest supports Common Streaming through the StreamingInfo element. This element provides information to provide Common Streaming information corresponding with an audio, video or subtitle track that is part of a DCC.

The Download Client SHALL download CMPs in accordance with the processes defined in this section.

# Content Fulfillment Specification Version 1.0

## 4.4.1 CMP Download and Update Process

This section defines the process for downloading and uploading CMPs.

Schema documentation conventions used this section follows conventions defined in [DMeta].

### 4.4.1.1 Download Process

CMP Download requirements are designed to support the following basic use cases:

- Download of a complete CMP that corresponds with a multi-track DCC. The Download Client downloads all tracks into a complete CMP that would correspond with a multi-track DCC.
- Download a subset of tracks into a CMP with the option of downloading some or all of any remaining tracks later. In this case, the OCMP is unchanged so the only download required is the new tracks.
- The OCMP has changed for any of a variety of reasons, such as track recall/replace, CMP part replacement (e.g., metadata) or the addition of new tracks. The Download Client would determine the changes, make the necessary updates, and download the additional elements as it determines appropriate. Note that a Download Client might discover an OCMP with a track recall change through a licensing failure.

The process always begins with an Initial Download. This includes the OCMP, other essential components, tracks indicated by the 'forceDownload' mechanism and any optional tracks the Download Client decides to download.

- Additional tracks can be downloaded following the original download in a process called Subsequent Download. In any instance where the OCMP might have been updated, it is necessary to check for OCMP updates prior to performing such a download. Note that certain licensing failures can indicate an OCMP update, so the Download Client might check for updates in this condition.

The process for Identifying Items that Require Download is as follows:

- DCC Parts to download are listed in CMPItem/OCMP/Presentation/APIID. If APIID/@forceDownload='true' the DCC with DCCResource/APIID matching Presentation/APIID must be downloaded.
- Application Parts to download are listed in CMPItem/OCMP/Application. If Application/@forceDownload='true' Parts referred to by CMPItem/AppResourceList/AppResource within a CMPItem/AppResourceList whose

## Content Fulfillment Specification Version 1.0

@ApplicationID attribute matches CMPItem/OCMP/Application/@ApplicationID must be downloaded if CMPItem/AppResourceList/AppResource@forceDownload='true'. The Application consists of items referenced in all instances of the associated CMPItem/AppResourceList/AppResource.

- Other Parts to download are listed in CMPItem/OCMP/OtherPart and CMPItem/OCMP/Presentation/DependentPart. If OtherPart/@forceDownload='true' the OtherPart matching CMPItem/OtherResource/@LocalSource must be downloaded. If DependentPart/@forceDownload='true' the DependentPart matching OtherResource/@LocalSource must be downloaded.
- If two Parts have the same LocalSource, it need not be downloaded more than once. This could occur, for example, if Presentations share DCCs or image Parts.

DCCResource, AppResource, OtherResource and OtherPart all have elements to support download. These elements are:

- LocationURL is the Part download link.
- Hash is the Part hash
- Length is the Part length
- LocalSource is the local filename. Some filenames are constrained as specified in [DDMP], Section 4.1.3.

### 4.4.1.2 Initial Download

The Download Client SHALL follow this process for initial downloading of a CMP:

- Retrieve Manifest from Download Server (as described above)
- Identify OCMP from Manifest's CMP element
- Download OCMP from Download Server using LocationURL
- Identify Items that Require Download (as defined above)
- Download (as defined above)

The Download Client SHOULD validate the OCMP and all Parts downloaded by calculating the hash and length of the downloaded OCMP or Part against the Part hash, if provided and Part length provided in in the Fulfillment Manifest.

# Content Fulfillment Specification Version 1.0

## 4.4.1.3 Subsequent Download

Subsequent Download is the capability to download portions of a CMP distinctly from Parts downloaded through the 'forceDownload' mechanism. It could in conjunction with the original download or it could be much later, perhaps years later. Subsequent Download includes the download of optional tracks (i.e., those not signaled as forced through the 'forceDownload' mechanism), track updates required due to a track recall, and other Parts of a CMP as required or desired.

It is not always necessary to download all items when the CMP is downloaded. That is, not all Parts are necessarily marked with `@forceDownload='true'`. In this case, the Download Client can download Parts later. Typically, the Download Client would determine which DCC or Application to download by using metadata information in the CMP, then obtain a new Manifest, locate the matching Part by APID or ApplicationID and download the associated Container or Application Parts

However, the potential exists for a DCC or Application to be replaced or recalled after the CMP was downloaded, outdating the information in the CMP.

The Manifest supports both downloading and updating any CMP Parts. The Manifest also provides information to clean up unused Parts.

### 4.4.1.3.1 DCC and Media Application Subsequent Download

DCC Subsequent Download is downloading a DCC Part of the CMP after the initial CMP Download is complete and, where applicable, adding the DCC to the CMP.

Media Application Subsequent Download is downloading a Media Application Part of the CMP after the initial CMP Download is complete and, where applicable, adding the Media Application to the CMP.

APIDs are identified by looking for Manifest/CMPItem/AppResource elements containing APID elements matching the desired APID. Once matched, the LocationURL can be used to download the Container.

The Download Client SHALL follow this process for subsequent downloading of a CMP:

- Retrieve Fulfillment Manifest from Download Server (as described above)
- Determine if any CMP Updates are required. If an update is required, perform a CMP Update and restart the entire process (i.e., retrieve a Fulfillment Manifest). Note that it is possible that the Updated CMP will not have the desired APID, so the Download Client might also need to restart whatever process initiated the Subsequent Download.
- If update is not required, proceed

## Content Fulfillment Specification Version 1.0

- Identify Parts that Require Download (as defined above)
- Download (as defined above) DCC or Media Application Parts.

The Download Client SHOULD validate all Parts downloaded by calculating the hash and length of the downloaded OCMP or Part against the Part hash, if provided and Part length provided in in the Fulfillment Manifest.

### 4.4.1.4 Update Process

In the time that passes between Initial Download and Subsequent Download, there may be updates to the CMP. For example, when a DCC is been replaced the Table Of Content will be updated to reference an updated Presentation Description referencing the new DCC, so the these Parts along with the DCC must be updated. The update process uses a combination of the [SMPTE2053] update process based on Version along with additional information provided in the Fulfillment Manifest. The Manifest also contains information the Download Client can use to remove Parts that are no longer needed.

The Download Client SHALL follow this process for update a CMP:

- SMPTE 2053 Part Update Process updates Table Of Contents Document and Presentation Description Documents. This process is as follows:
  - The Download Client determines an update is necessary by comparing the Version in the CMP's TableOfContent/@Version with the Version in CMPItem/UpdatedTOC/@Version. If CMPItem/UpdatedTOC is not present, no update is necessary.
  - If UpdatedTOC/@Version is greater that TableOfContents/@Version, TableOfContents must be updated with the Part referenced by UpdatedTOC.
  - Presentation Description Documents are then updated in accordance with [SMPTE2053].
- The DCC Update process updates DCCs. This process is as follows:
  - Note that the function of the DCC is to update DCCs that are already in the CMP. Other DCCs referenced in OCMP and Presentation Description Documents can be ignored. Note that OCMP and Presentation Description Documents to not relate old DCCs to new DCCs so it is impractical, although possible, to use these objects for update.
  - For each DCC in the CMP, the APID of that DCC is checked against ObsoleteAPID elements in DCCResource. If a match is found the downloaded DCC is obsolete and its replacement is in referenced in the DCCResource element that includes the match.

## Content Fulfillment Specification Version 1.0

- DCCs are then updated with the referenced DCCs.

Update information structure in the CMP is defined in [DDMP], Section 4.3.1.

If the Download Client opts to download a DCC or Media Application, then the Download SHALL perform the CMP update functions.

### 4.4.2 CMP File Maintenance

When files are replaced, they can waste space in the CMP. In most circumstances it is clear which files are no longer needed. These can be removed prior to downloading the updates, with the understanding that a failed download might result in an incomplete CMP (e.g., Presentation part failed to download).

The ZIP file structure allows for gaps in the ZIP where files used to be; however, it is beneficial to avoid internal fragmentation in the ZIP by filling empty spaces with subsequent downloads. Implementers can support more advanced internal fragmentation management, including moving files, if they choose. This is at the discretion of the Client Implementer.

The Fulfillment Manifest includes information about obsolete Parts in the form of ObsoleteAPID and ObsoleteLocaSource elements.

Download Clients SHOULD remove Obsolete Parts from the CMP.

# Content Fulfillment Specification Version 1.0

## 5 Fulfillment Manifest File and element

This section defines the Fulfillment Manifest File and FulfillmentManifest element.

### 5.1 FulfillmentManifest File

The Fulfillment Manifest is returned as a file containing an XML Document consisting of a FulfillmentManifest element.

### 5.2 FulfillmentManifest Element

| Element             | Attribute | Definition   | Value                           | Card. |
|---------------------|-----------|--|---------------------------------|-------|
| FulfillmentManifest |           | The root level element of a Fulfillment Manifest file. | fmdece:FulfillmentManifest-type |       |

#### 5.2.1 FulfillmentManifest-type

CMP support items are distinct from non-CMP items to ensure backwards compatibility. Item is not used for DCCs within a CMP; for that, CMPItem/DCCResource is used.

| Element                         | Attribute | Definition  | Value                                     | Card. |
|---------------------------------|-----------|---|---|-------|
| <b>FulfillmentManifest-type</b> |           |   |   |       |
| ALID                            |           | Asset Logical ID fulfilled by this Manifest                               | md:AssetLogicalID-type                    |       |
| CMPItem                         |           | Information about the CMP   | fmdece:FulfillmentManifestCMP-type        | 0..n  |
| StreamingInfo                   |           | Information to support Common Streaming of Content. One instance per APID | fmdece:FulfillmentManifestStreamInfo-type | 0..n  |

#### 5.2.2 FulfillmentManifestCMP-type

This element contains information necessary to download the CMP and the related parts, including Presentations (including DCCs), Applications and other components.

## Content Fulfillment Specification Version 1.0

| Element                            | Attribute | Definition  | Value  | Card. |
|------------------------------------|-----------|---|--|-------|
| <b>FulfillmentManifestCMP-type</b> |           |   |  |       |
| Description                        |           | Description of the CMP. This is provided for user interfaces  | xs:string                                      | 1..n  |
|                                    | Language  | Language associated with Description. At most one instance of Description may omit Language. If missing, the associated Description is the default. | xs:language                                    | 0..1  |
| LocationURL                        |           | URL reference to location(s) of OCOMP. May include access control information. If multiple instances exist, any may be used.                        | xs:anyURI                                      | 1..n  |
| Hash                               |           | File Hash   | md:Hash-type                                   | 0..1  |
| Length                             |           | Byte length of the file   | xs:integer                                     |       |
| LocalName                          |           | Name for CMP in local file system.  | xs:string                                      | 0..1  |
| OCMP                               |           | Definition of the CMP   | mddece:OCMP-type                               |       |
| DCCResource                        |           | Information about DCCs in the CMP   | fmdece:FulfillmentManifestDCCResource-type     | 0..n  |
| AppResourceList                    |           | Information about Media Applications in the CMP   | fmdece:FulfillmentManifestAppResourceList-type | 0..n  |
| OtherResource                      |           | Information about other resource in the CMP. For example, BaseLocations.  | fmdece:FulfillmentManifestOtherResource-type   | 0..n  |
| UpdatedTOC                         |           | Provides information about Table Of Content replacements, if any.<br>UpdatedTOC/Type='TableOfContents'  | fmdece:FulfillmentManifest2053Part-type        | 0..1  |

# Content Fulfillment Specification Version 1.0

## 5.2.3 Types used to define CMP Download and Update

There are four primary types of CMP Part that are individually addressed in the Fulfillment Manifest

- DCC Resource – Part of a Presentation, these Resources are DCCs. Each DCCResource element refers to a DCC Resource.
- Application Resources – A Media Application contains one or more Application Resources. AppResourceList is a collection of all Application Resources associated with an ApplicationID. AppResourceList/AppResource refers to an Application Resource.
- Other Resources – Other Resources can be included in a CMP. At the top level is BaseLocations. Other Resources can include CMP Parts or Presentation Parts that are not DCCs, Applications or [SMPTE2053] parts (i.e., TOC, Presentation Description Document and Media Application Document). OtherResource elements refer to other Resources.
- Table of Contents – The [SMPTE2053] TOC is required for a OCMF. However, the TOC can be updated. UpdatedTOC, if present, references the most recent TOC.

For each of these types an element type is defined to provide information on downloading, updating and removing, as appropriate, Resources of that type.

### 5.2.3.1 FulfillmentManifestDCCResource-type

This element defines how to download the DCC associated with a given APID.

A DCC is obsolete if its APID matches an instance of DCCResource/ObsoleteAPIID.

| Element                                 | Attribute | Definition   | Value  | Card. |
|---|-----------|--|--|-------|
| FulfillmentManifest<br>DCCResource-type |           |  | fmdece:FulfillmentManifestResourceRef-type<br>(by extension) |       |
| APID                                    |           | APID of the DCC  | md:AssetPhysicalID-type                                      |       |
| ObsoleteAPIID                           |           | APID of DCCs that have<br>been replaced by this<br>DCC | md:AssetPhysicalID-type                                      | 0..n  |

### 5.2.3.2 FulfillmentManifestAppResourceList-type

An Application consists of one or more Application Resources referenced by AppResource elements. Within a given AppResource element, active Application Resources are listed in

## Content Fulfillment Specification Version 1.0

AppResourceList/AppResource elements. Note that this refers to any Media Application Resource, whether it is referenced by a Presentation or the TOC.

If an Application Resource is obsolete its filename is listed in AppResourceList/ObsoleteLocalSource.

| Element                                 | Attribute     | Definition  | Value   | Card. |
|---|---------------|---|---|-------|
| FulfillmentManifestAppResourceList-type |               |   |   |       |
|   | ApplicationID | Application ID for the Application  | md:id-type  |       |
| AppResource                             |               | Information required to download or update an Application Resource.   | fmdece:FulfillmentManifestResourceRef-type (by extension) | 1..n  |
|   | isPresent     | Indicates whether the Application Resource is present in the OCMP.  |   |       |
|   | forceDownload | Indicates whether the Download Client must download the Application Resource immediately following OCMP download. |   |       |
|   | overwrite     | Indicates whether file of same LocalSource should be overwritten.   |   |       |
| ObsoleteLocalSource                     |               | LocalSource of Resources replaced by this Resource.   | xs:string   | 0..n  |

### 5.2.3.3 FulfillmentManifestOtherResource-type

Other Resources include BaseLocations, Image files and metadata. Some of these files are included in the OCMP, so they only will be included in the Manifest if they have been updated.

Some types of Resources have the own versioning mechanism. If so, the means of determining replacement is defined in [DDMP], Section 4.3.1. Otherwise, Resources are determined to be obsolete if their filename including path matches OtherResource/ObsoleteLocalSource.

## Content Fulfillment Specification Version 1.0

| Element                                       | Attribute | Definition  | Value  | Card. |
|---|-----------|---|--|-------|
| FulfillmentManifest<br>OtherResource-<br>type |           |   | fmdece:FulfillmentManifestRe<br>sourceRef-type<br>(by extension) |       |
| Type  |           | Type of this Resource   | fmdece:FulfillmentManifestTy<br>pe-type                          |       |
| Version                                       |           | Version of the Resource.<br>The meaning of Version<br>depends on the type of<br>Resource. | xs:string  | 0..1  |
| ObsoleteLocalSour<br>ce                       |           | LocalSource of Resources<br>replaced by this<br>Resource.                                 | xs:string  | 0..n  |

Hash SHALL be included.

The Type element SHALL have one of the following values

- 'Image' – Image file, compatible with [DMeta], Section 3.2.
- 'BaseLocation' – BaseLocations document as per [DMeta], Section 4.4.
- 'Other' – File that is not one of the above.

### 5.2.3.4 FulfillmentManifest2053Part-type

Each instance of this element describes a part in the CMP subject to replacement. If @Version is more current than Version in the CMP, this part is more current.

Any [SMPT2053] Presentation Part or Application Part that is not referenced by TableOfContents is obsolete.

| Element                              | Attribute | Definition | Value  | Card. |
|--------------------------------------|-----------|------------|--|-------|
| FulfillmentManifest<br>2053Part-type |           |            | fmdece:FulfillmentManifestRe<br>sourceRef-type<br>(by extension) |       |

## Content Fulfillment Specification Version 1.0

| Element | Attribute | Definition   | Value                 | Card. |
|---------|-----------|--|-----------------------|-------|
|         | Version   | Version of the file referenced here. Version corresponds with Version definitions in [SMPTE2053] | xs:nonNegativeInteger |       |
| Type    |           | Type of Part (see below)   | xs:string             | 0..1  |

### 5.2.3.5 FulfillmentManifestResourceType-type

This simple type is xs:string and SHALL have one of the following values:

- 'TableOfContents' – [SMPTE2053] Table Of Contents
- 'PresentationDescription' – [SMPTE2053] Presentation Description document (//Presentation)
- 'MetadataMovie' – MetadataMovie document as per [DDMP]
- 'MetadataTail' – MetadataTail document as per [DDMP]
- 'Image' – Image document as per [DDMP]
- 'Other' – File that is not one of the above.

### 5.2.4 FulfillmentManifestStreamInfo-type

This element provides streaming information corresponding with a Physical Asset, reference by APID. The principle is that an Asset might be represented by a downloadable DCC associated with that APID and one or more Common Streaming forms associated with that same APID.

As a DCC can contain more than one track, the TrackReference attribute disambiguates which track is described by this element.

| Element                            | Attribute      | Definition   | Value                   | Card. |
|------------------------------------|----------------|--|-------------------------|-------|
| FulfillmentManifestStreamInfo-type |                |  |                         |       |
|                                    | PresentationID | The PresentationID for the Media Presentation that can be streamed through this MPD. | md:PhysicalAssetID-type |       |

## Content Fulfillment Specification Version 1.0

| Element                     | Attribute | Definition  | Value                              | Card. |
|-----------------------------|-----------|---|------------------------------------|-------|
| MPDLocation                 |           | URL representing location of MPD. If multiple instances exist, any may be used.           | xs:anyURI                          | 1..n  |
| ExperienceMedia Application |           | Information to select titles and tracks, and otherwise navigate the streaming experience. | mddece:ExperienceStreamApp-type    | 0..1  |
| BaseLocations               |           | BaseLocation information required to acquire licenses.                                    | mddece:ContainerBaseLocations-type | 0..1  |

### 5.2.4.1 FulfillmentManifestResourceRef-type

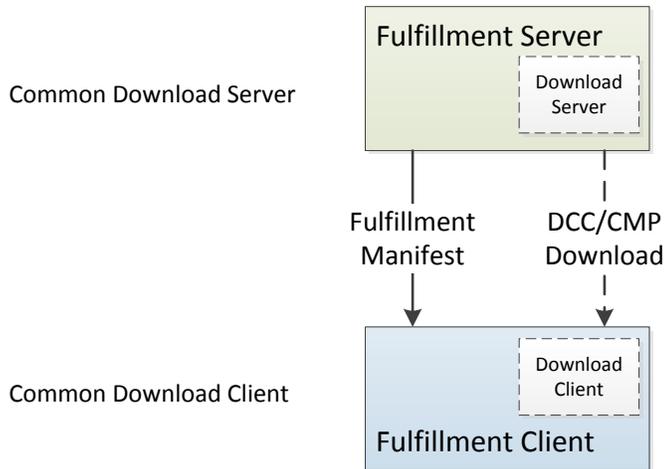
This complex type is used to refer to Resources. It is a building block for other types that reference Resources.

| Element                              | Attribute | Definition   | Value        | Card. |
|--------------------------------------|-----------|--|--------------|-------|
| FulfillmentManifest ResourceRef-type |           |  |              |       |
| LocationURL                          |           | URL reference to location(s) of Container. May include access control information. If multiple instances exist, any may be used. | xs:anyURI    | 1..n  |
| Hash                                 |           | File hash  | md:Hash-type | 0..1  |
| Length                               |           | Byte length of the file  | xs:integer   |       |
| LocalSource                          |           | Part Name in accordance with [SMPTE2053]   |              |       |

# Content Fulfillment Specification Version 1.0

## 6 Common Download Client and Server

This section defines the requirement for a Common Download Client and Common Download Server. These generic entities are intended to be referenced as a core set of requirements for an ecosystem-specific Download Clients and Servers. This is illustrated as the following subset of the diagram in Section 2:



### 6.1 Common Download Server Requirements

A Common Download Server SHALL comply with the requirements established for a Fulfillment Server in Section 3.

A Common Download Server SHALL comply with requirements established for a Download Server in Section 3.

### 6.2 Common Download Client Requirements

A Common Download Client SHALL comply with requirements established for Fulfillment Client in Section 4.

A Common Download Client SHALL comply with requirements established for Download Client in Section 4.

### END ###