DECE Media Package (DMP) Specification

Version 1.0r1 14 November 2013

DECE Media Package Specification Version 1.0r1

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 23 November 2013.

Optional Implementation Agreement:

DECE offers an optional license agreement relating to the implementation of this document. A copy is available from DECE upon request. Entities executing the 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.

Contact Information:

Licensing and contract inquiries and requests should be addressed to us at:
http://www.uvvu.com/uv-for-business.php

The URL for the DECE web site is http://www.uvvu.com

DECE Media Package Specification Version 1.0r1

Contents

1 Introduction 4
1.1 Scope 4
1.2 Document Organization 4
1.3 Document Notation and Conventions 4
1.4 Normative References 4
1.4.1 DECE Normative References 4
1.4.2 External References 5
1.5 Informative References 6
1.6 Terms, Definitions and Acronyms 6
1.7 XML Change Management 6
2 DECE Media Packages and DECE Ecosystem (Informative) 7
3 DECE Media Package Overview 9
3.1 Overview of DMP 9
3.2 DMP Use of SMPTE Media Package 10
4 DECE Media Package Contents 12
4.1 DMP Structure 12
4.2 DMP Zip File Constraints 13
4.3 DMP Parts 16
4.3.1 Versioning, Naming and Types 16
4.3.2 SMPTE Media Package objects 20
4.3.3 Media 23
4.3.4 Metadata and Images 23
4.3.5 Base Locations and Licenses 24
4.3.6 Constraints on DCCs within DMPs 25
4.3.7 Media Applications 25

1 Introduction

1.1 Scope

This document specifies a format for DECE Media Packages (DMP).

1.2 Document Organization

This document is organized as follows:

1. Introduction—Provides background, scope and conventions
2. DECE Media Package and DECE Ecosystem
3. DECE Media Package Overview
4. DECE Media Package Contents

1.3 Document Notation and Conventions

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]. For more information, please refer to those directives.

- 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.

1.4 Normative References

1.4.1 DECE Normative References

The following DECE technical specifications are cited within the normative language of this document.

[DSystem]	System Specification
[DMeta]	Content Metadata Specification
[DMedia]	Common File Format& Media Format Specification

1.4.2 External References

The following external references are cited within the normative language of this document.

[DASH]	ISO/IEC 23009-1:2011, "Dynamic Adaptive Streaming over HTTP"
[SMPTE2053]	SMPTE ST 2053:2011, *Media Package for Storage, Distribution and Playback of Multimedia File Sets and Internet Resources*, July 13, 2011.
[ISO14496-12]	ISO/IEC 14496-12, Third Edition, "Information technology — Coding of audio-visual objects – Part 12: ISO Base Media File Format", including Amendment 3 and prior amendments and corrigenda
[ISO29500-2]	ISO/IEC 29500-2 *Open Packaging Conventions* (OPC, 2008 November 15), http://standards.iso.org/ittf/PubliclyAvailableStandards/c051459_ISOIEC%2029500-2_2008%28E%29.zip
[UNICODE]	UNICODE 6.0.0, "The Unicode Standard Version 6.0", http://www.unicode.org/versions/Unicode6.0.0/
[TR-META-CM]	*Common Metadata,* TR-META-CM, v2.1b, July 12, 2013, Motion Picture Laboratories, Inc., http://www.movielabs.com/md/md/v2.1/Common_Metadata_v2.1b.pdf
[XSD-META-CM]	XML Schema to accompany [TR-META-CM], July 12, 2013, http://www.movielabs.com/schema/md/v2.1/md-v2.1.xsd
[ISO-P2H]	ISO/IEC Directives, Part 2, Annex H http://www.iec.ch/tiss/iec/Directives-part2-Ed5.pdf
[RFC4122]	Leach, P., et al, A Universally Unique IDentifier (UUID) URN Namespace, July 2005 http://www.ietf.org/rfc/rfc4122.txt

Note: Readers are encouraged to investigate the most recent publications for their applicability.

1.5 Informative References

The following external references are cited within the informative language of this document.

[DPublisher]	DECE Content Publishing Specification, Version 1.0.3
[CENC]	ISO/IEC 23001-7:2012, First edition 2012-02-01, "Information technology - MPEG systems technologies - Part 7: Common encryption in ISO base media file format files"

1.6 Terms, Definitions and Acronyms

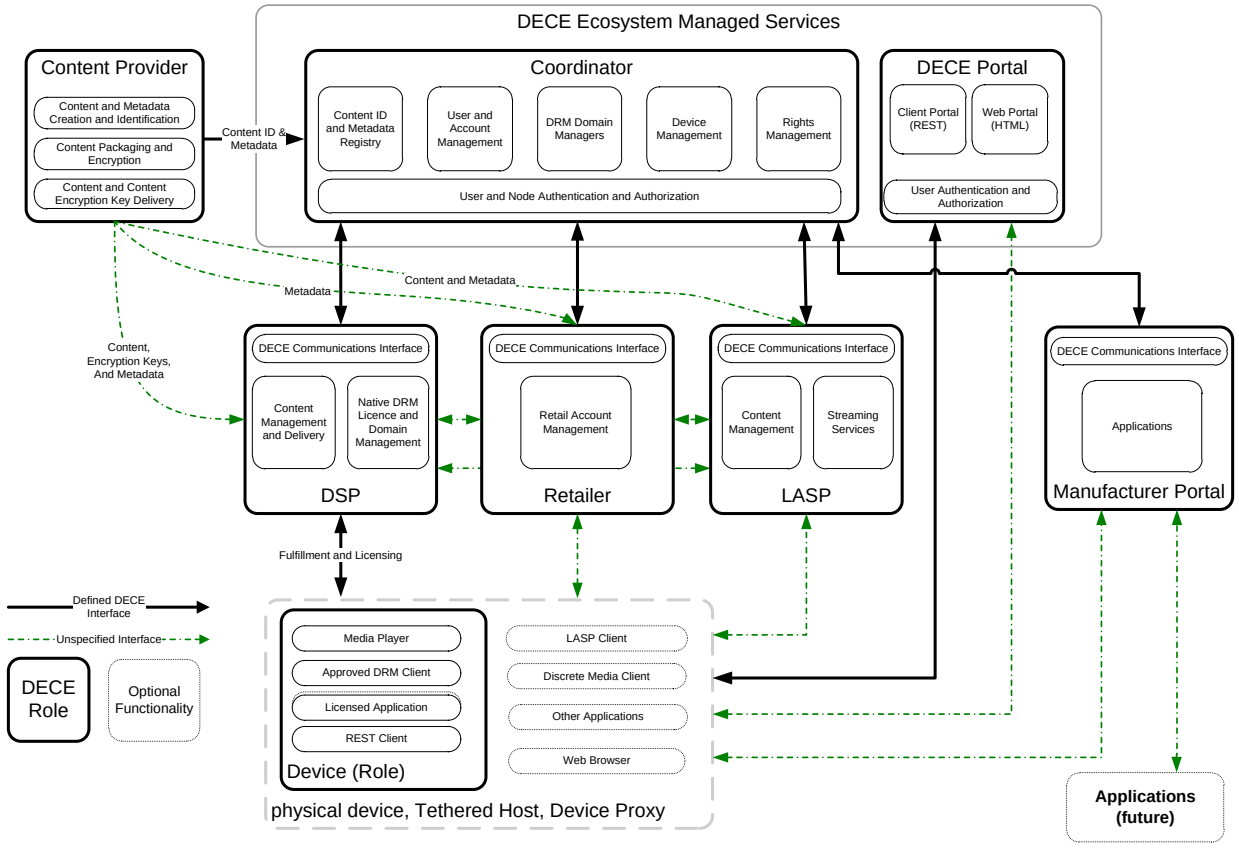
DECE Media Package terminology is defined in [DSystem], Section 1.4.

1.7 XML Change Management

Recipients of XML Documents encoded using this specification SHALL comply with XML Change Management defined in [DSystem], Section 1.6.

2 DECE Media Packages and DECE Ecosystem (Informative)

DECE Media Packages are delivered from Content Providers to DSPs as shown in the following diagram as "Content". Also, although not specifically shown, Content Providers can also deliver DMPs to LASPs.



Original DMPs (ODMPs) are created by Content Providers analogously to ODCCs. DMPs are delivered to Devices analogously to DCCs. New processes include Late Binding, the process of collecting multiple DCCs associated with a single Media Presentation into a DMP; and Common Streaming, the process of streaming Common Streaming Format DCCs. Devices can simultaneously play stored tracks (DMP) formats and streaming tracks (Common Streaming).

In its simplest form, the DMP is functionally equivalent to a single DCC. It contains ZIP components, such as a directory; SMPTE Media Package components, such as a Table of Contents; and a DCC. All Devices can locate the DCC within the DMP and play it just as if the DCC were not in a DMP.

A DMP can contain more than one DCC. All Devices can recognize and play multi-track DCCs based on the Common File Format as defined in [DMedia]. Devices that support Late Binding are capable of playing tracks from multi-track DCCs that are built accordingly to [DMedia], Sections B.7 and C.7.

It is also possible to construct and deliver a DMP that has no media Containers. The DMP is built with the same components, less the DCCs. There is enough information in the DMP to determine what pieces are missing so they can be obtained. To support this functionality, information that typically resides in the DCC is replicated in the DMP, outside of the DCC. For example, metadata that is required for the DCC must exist in the DMP, whether or not the DCC is present. Similarly, the DMP supports the storage of licenses and data such as BaseLocation and Purchase URL (PURL) references.

DMPs also support the inclusion of Media Applications, such as applications to navigate Media Presentations and, ultimately, more advance applications such as games.

3 DECE Media Package Overview

3.1 Overview of DMP

The DECE Media Package (DMP) is an application of the SMPTE Media Package standard (ST2053) with constraints and requirements specific to storage, download, and playback of CFF files and CFF Track Files.

At minimum, DMPs consist of a Zip container conforming to ISO/IEC 29500-2 that contains XML manifest files identified as one Table of Contents document, and one or more Presentation Description documents. DMPs may also contain media and other files. Files do not use Zip compression, so they may be read in-place, without extraction, similar to a directory or folder in the device's native file system.

DMPs enable "on demand" download and storage of media files, metadata files, presentation applications, licenses, and other files identified by APID and downloaded using URLs listed for each APID in a current download manifest. DMP manifest files contain version numbers so that new manifest files, media files, and other files that have been updated or added to a presentation can be downloaded by a device to update its locally stored DMP. This update operation only requires the client to add or replace files in the DMP Zip container, not to edit XML manifest documents, which are updated by the publisher when new or different content is offered for that package.

Publishers can prepackage files that are most likely to be used, and make other files available for download on demand. When other audio tracks, subtitle tracks, presentation metadata or presentation applications become available later, publishers can update their Web resident manifest files and download manifest to make the new files available for download on demand. A user who downloads files to a DMP may copy that DMP as a single Zip file to their other DECE Devices, retaining all the files that have been downloaded.

Devices can identify compatible presentations and tracks for offline late bound playback by reading the Table of Contents and Presentation Description manifest files, and reading the DMP Zip directory to identify APID file names corresponding to Presentation Description tracks and resource files that are stored in the DMP. When online, users have the option to download or progressive download files corresponding to APIDs listed in manifest files that are not yet downloaded and present in the Zip directory.

3.2 DMP Use of SMPTE Media Package

A SMPTE Media Package is a Zip container derived from the ISO/IEC 29500-2 Open Packaging Convention for the purpose of storing media presentations, and it additionally specifies manifest files consisting of a Table of Contents XML document, and one or more Presentation Description XML documents defined in the SMPTE ST2053 standard. Media files and related manifest and presentation files may be prepackaged in the Zip container, stored as "Parts[1]", or all except the TOC file may be downloaded on demand to add them to a locally stored the Media Package.

Version numbers identify newer manifest files and media files available as Web resources that may be downloaded to replace older files in a local Media Package, and new files, such as CFF Track Files, may be added to new manifest versions at any time to make them available for download.

An important DMP constraint on SMPTE Media Package is that DMP SHALL exclude the OPC specified "/rels/.rels" package-level Relationships part and associated "/[Content_Types].xml" file as defined in ISO 29500-2, Section 10.2.61 that specifies the MIME Media Types for all default file extensions and all parts with non-default file extensions that are stored in the package.

The reason for this exclusion is to simplify DMP media players by only requiring that they add files using basic Zip functionality without having to edit XML in the /.rels part and /[Content_Types].xml document when files are added. This constraint requires players to check the Zip directory to determine if files are currently stored in the Zip container, and relies on a naming convention to associate filenames in Presentation Description RemoteSource elements with filenames in the Zip directory and URLs in the Download Manifest (defined in [DSystem] Section 1.4). DMP SHALL ignore the LocalSource element since it cannot be resolved through /.rels.

Below is an overview of the SMPTE Media Package XML structure as applied to DMP. The section that follows this one specifies the mapping between DECE defined metadata to SMPTE manifest attributes and elements.

[1] Parts is an [ISO29500-2] term) referring to the files within the ZIP container associated with a particular function

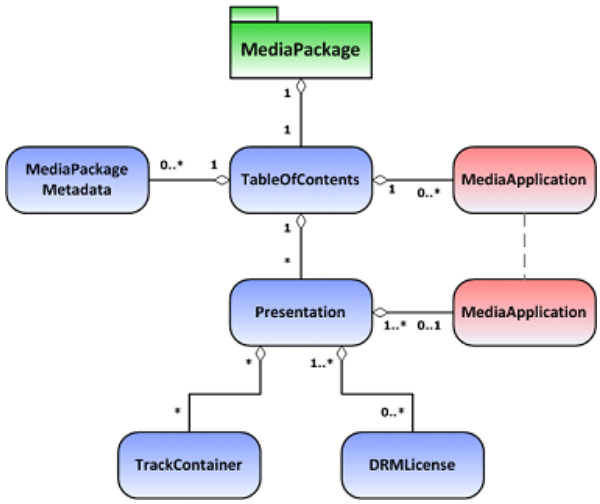
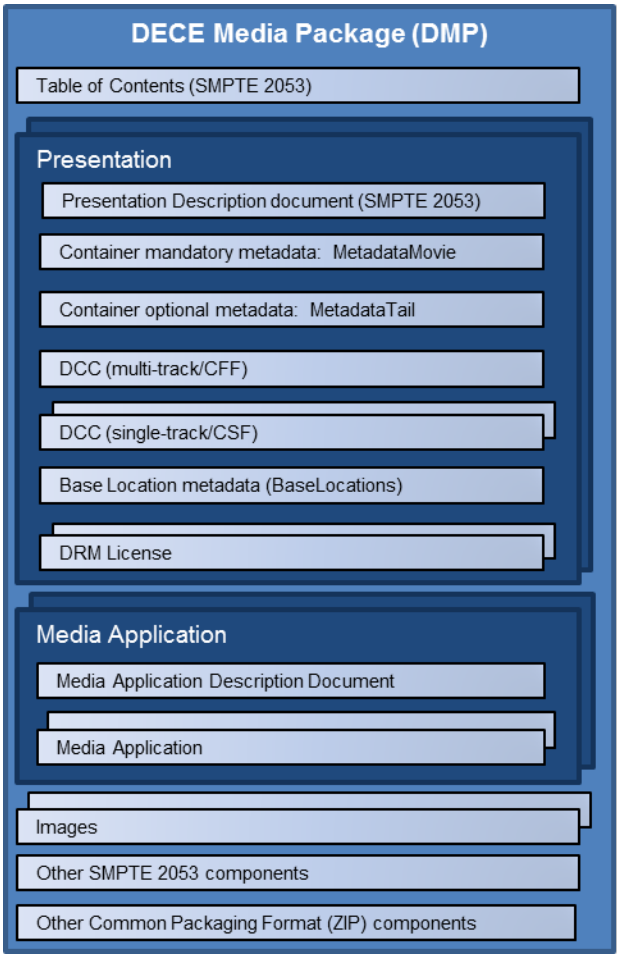


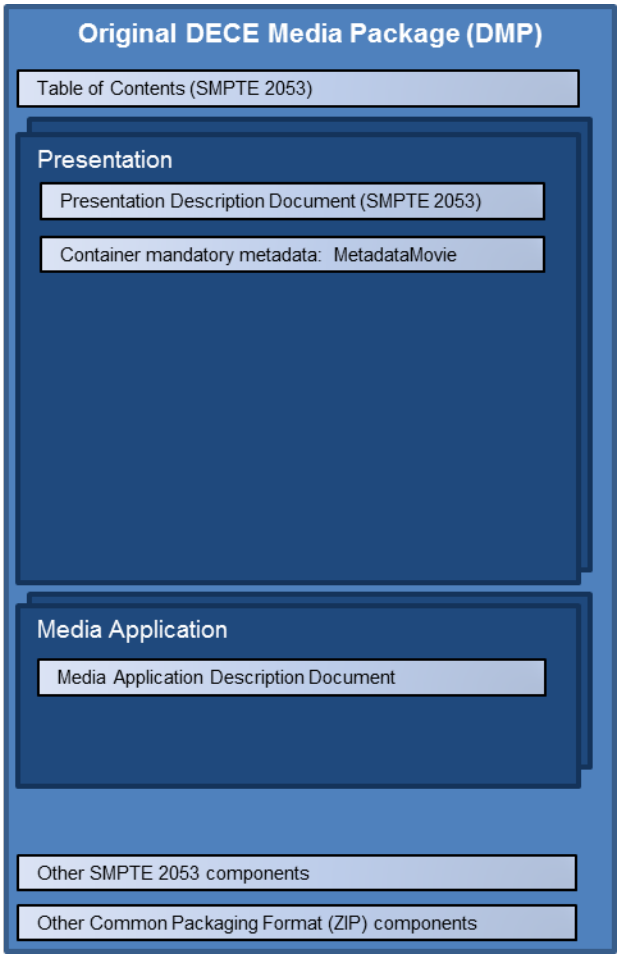
Figure 3-1 SMPTE Media Package structure, showing components unique to a Media Package

4 DECE Media Package Contents

A DMP contains ZIP components such as a director, and a series of files. This section defines those files and when those files are required.

The following illustrates the components of DMP and an ODMP. This is a logical representation and components could be positioned differently. The ODMP illustration shows the necessary components, although any other component can be published in an ODMP.





4.1 DMP Structure

DMP SHALL comply with [SMPTE2053], except when noted.

4.2 DMP Zip File Constraints

Where requirements conflict with Open Packaging Convention [ISO29500-2], this document takes precedence.

The DMP SHALL NOT use ZIP encryption.

Normative language is as follows:

- 'SHALL be = "<value> (or similar) indicates that the field is required in the DMP and must be set to <value>
- 'SHALL be included' indicates the item is required in the DMP with any constraints as indicated. Any decoding device must be capable of properly decoding the DMP with the values set accordingly.
- 'SHOULD/MAY be included' indicates the item is optionally included in the DMP, but only with the constraints given. The item SHALL not be included in a manner that conflicts with the constraints, unless otherwise noted. Any decoding device must be capable of properly decoding the DMP with the values set accordingly.
- "MAY ignore" indicates the items is optionally included in the DMP. Any entity decoding device can safely ignore this item.

DMPs SHALL comply with the following constraints:

Local File Header:

local file header signature	SHALL be = 0x04034b50
version needed to extract	SHALL support 1.0 as baseline, 4.5 for Zip64, 6.3 for UTF-8 *[Note 1]*
general purpose bit flag	
bit 0 (encryption)	SHALL be = 0 (not encrypted)
bits 1-2	MAY ignore
bit 3 (CRC and size values follow file data)	SHALL be = 1
bits 4-10	MAY ignore
bit 11	SHALL be included
bit 12-15	MAY ignore
compression method	SHALL be = 0 (no compression)

last mod file time last mod file date (above)	Was "MAY Ignore" I think it should be "SHALL be included. What a Part is updated, it SHALL be of a date-time later than all previous versions."
crc-32	SHALL be included
compressed size	SHALL be included, supporting both 4- and 8-byte fields
uncompressed size	SHALL be included, supporting both 4-and 8-byte fields
file name length	SHALL be included
extra field length	SHALL be included
file name (variable size)	SHALL be included
extra field (variable size)	SHALL support ID 0x0001 (zip64), MAY ignore others

Note 1: Only the Zip64 feature of ZIP 4.5 and the UTF-8 encoding feature of ZIP 6.3 are required.

File data SHALL be supported.

Data descriptor SHALL be supported.

Archive decryption header MAY be ignored.

Archive extra data record MAY be ignored.

Central Directory File Header (same as Local File Header except):

central file header signature	SHALL be = 0x02014b50
version made by	MAY ignore
file comment length	SHALL be supported
disk number start	SHALL be = 0
internal file attributes	MAY ignore
external file attributes	MAY ignore
relative offset of local header	SHALL be supported
file name (variable size)	SHALL be supported
extra field (variable size)	SHALL support ID 0x0001 (zip64), MAY ignore others
file comment (variable size)	MAY ignore

The Central Directory SHALL NOT be encrypted.

The Central Directory SHALL NOT be compressed.

Zip64 end of central directory record (same as Local File Header except):

zip64 end of central dir signature	SHALL be = 0x06064b50
size of zip64 end of central directory record	SHALL be supported
number of this disk	SHALL be = 0
number of the disk with the start of the central directory	SHALL be = 0
total number of entries in the central directory on this disk	MAY ignore
total number of entries in the central directory	MAY ignore
size of the central directory	MAY ignore
offset of start of central directory with respect to the starting disk number	MAY ignore
zip64 extensible data sector	MAY ignore

Zip64 end of central directory locator:

zip64 end of central dir locator signature	SHALL be = (0x07064b50)
number of the disk with the start of the zip64 end of central directory	SHALL be = 0
relative offset of the zip64 end of central directory record	SHALL be supported
total number of disks	SHALL be = 1

End of central directory record:

end of central dir signature	SHALL verify = (0x06054b50)
number of this disk	SHALL support if 0
number of the disk with the start of the central directory	SHALL support fi 0
total number of entries in the central directory on this disk	MAY ignore
total number of entries in the central directory	MAY ignore
size of the central directory	MAY ignore
offset of start of central directory with respect to the starting disk number	SHALL support

.ZIP file comment length	SHALL support
.ZIP file comment	MAY ignore

4.3 DMP Parts

In the following statements, the terms 'Part' and 'Parts' (an [ISO29500-2] term) refer to the files associated with a particular function. For example, when it is stated that a [SMPTE2053] Table of Contents is required, implicit in this statement is that, as required in [SMPTE2053].

The DMP SHALL contain all data required by [SMPTE2053], including data required by [ISO29500-2], unless otherwise noted.

For the avoidance of doubt, this includes both files and data within those files. All required XML document parts and attributes are required. Note that the DMP does not use the "/rels/.rels" and "/[Content_Types].xml" mechanisms, so those files are not required.

Unless otherwise noted, and with the exception of DECE backwards compatibility guidelines ([DSystem], 1.6], XML documents SHALL validate against the respective schemas associated with the same namespace.

The following terminology is used in this section

- 'Potential Media Presentation' refers to a Media Presentation that refers to a Media Presentation that could be included in the DMP. DCCs may be present or not. This is to distinguish from Media Presentations that are not supported by the Table of Contents, Metadata and/or other material.
- 'Populated Media Presentation" refers to a Media Presentation whose DCCs are included in the DMP.
- 'Unpopulated Media Presentation' refers to a Media Presentation whose DCCs are not included in the DMP.

In the following sections, XML definitions of [SMPTE2053] elements and attributes are defaulted to their definition in [SMPTE2053].

4.3.1 Versioning, Naming and Types

This section describes versioning and naming within a DMP.

4.3.1.1 Versioning

There are three mechanisms for tracking versions within a DMP

- Version elements and attributes. Some Parts contain information about their version. These are called Versioned Parts.
- Referenced media and applications. DCCs and Media Applications are identified by APIDs and AppIDs respectively. The correct version of DCCs and Media Applications are defined in the [SMPTE2053] Presentation Description documents and Media Application Description documents.
- Resource Name. Some Parts' versions are identified by unique names for each version. There is no information in the DMP that indicates which is the most current version, but external information can indicate which version of the Part is most current. These are called Resource Parts. Everything that is not a Version Part or Referenced media and applications is a Resource Part. From the standpoint of updates, the means of identifying a version of a Resource Part is the Part name (filename).

Versioned Parts include the following:

- Table of Contents document
- Presentation Description document
- Media Application Description document
- Container Mandatory Metadata
- Container Optional Metadata

XML documents defined by [SMPTE2053] contain a Version attribute. For documents other than TableOfContents, there is a corresponding RequiredVersion attribute corresponding referring XML documents. The principle is that that each XML object in the DMP has a Version that is greater than or equal to the corresponding Version attribute in referring document. For example, Presentation/@Version in a Presentation Description Document should be greater or equal to PresentationRef/@VersionRequired in the TableOfContents.

A DMP SHALL NOT contains XML documents whose Version attribute is less than the corresponding @VersionRequired attribute in the referring document.

Version applies to certain other Parts of the DMP.

In Container Required Metadata, MetadataMovie/@MetadataVersionReference is the 'Version' corresponding with VersionRequired attribute in referencing objects.

In Digital Asset Metadata, DigitalAsset/@updatenum is the 'Version' corresponding with VersionRequired in referencing objects.

Any DCC or Media Application Parts referenced in the [SMPTE2053] Presentation Description documents and Media Application Description documents is assumed to be current. Any DCC or Media Application Part not referenced is assumed to be obsolete.

Parts defined as Resources, that is Parts identified by Presentation/ResourceLibrary/Resource, have a unique name. This is included in Presentation/ResourceLibrary/Resource/@LocalSource and is the filename in the DMP. There is no information in the Part itself that indicates the Version. Version can be provided externally. Note that [DSystem], Section 11 defines a Manifest that includes Resource versioning information.

4.3.1.2 Naming and Types

Open Packaging Conventions identifies data by Parts, generally corresponding with files in a ZIP.

References in DMPs SHALL be Open Packaging Convention Part names. The OPC Relationships mechanism is not used. The SMPTE Media Package "/rels/.rels" is not part of a DMP.

All Part names SHALL use the Part URI format as per [ISO29500-2], Section 9. That is, certain characters require percent encoding.

Part names SHOULD NOT use characters that require percent encoding.

DMPs do not use the "[Content_Types].xml" mechanism of OPC, so therefore, unlike SMP, this is not required.

DMPs NEED NOT include "/[Content_Types].xml".

Naming SHALL be as shown in the following table

UTF-8 encoding of DCC filenames and comment fields is allowed.

Part	Name	Example
TOC	/TableOfContents.xml	`/TableOfContents.xml`
Presentation	/<PresentationID>/Presentation.xml	`/urn:dece:presentationid:org:craig:1235/Presentation.xml`
BaseLocations	/<PresentationID>/BaseLocations.xml	`/urn:dece:presentationid:org:craig:1235/BaseLocations.xml`

MetadataMovie	/<PresentationID>/MetadataMovie.xml	/urn:dece:presentationid:org:craig:1235/MetadataMovie.xml
MetadataTail	/<PresentationID>/MetadataTail.xml	/urn:dece:presentationid:org:craig:1235/MetadataTail.xml
Other Resources unique to Presentation	/<PresentationID>/<PresResource>	/urn:dece:presentationid:org:craig:1235/OtherData.xml
DCC	/<APID>.[uvu|uva|uvv|uvt]	/urn:dece:apid:eidr-x:50A5-34E1-4FFF-0BBD-17C9-G:1.uvu
MediaApplication	/<MediaApplicationID>/<AppName>	/urn:dece:applicationid:org:craig:1235/Experience.html
Media Application components???	/<MediaApplicationID>/<AppResource>	/urn:dece:applicationid:org:craig:1235/SubMenu.html
Images exclusive to Presentation	/<PresentationID>/<Image URN>	urn:dece:presentationid:org:craig:1235/ urn:dece:container:metadataimageindex:23.png
Shared images	/Images/<image name>	/Images/prettypicture.jpg
License	/Licenses/<DRM>/[<Presentation ID>|<APID>].uvl	/Licenses/playready/ urn:dece:presentationid:org:craig:1235.uvl
Other Resources shared between Presentations	/Resource/<resourcefilename>	/Resources/YetAnotherFile.xml

In a DMP, the LocalSource attribute in SMPTE Media Package XML Documents refers to an OPC Part. Since Part names are ZIP file names, LocalSource contains the file name within the ZIP.

The following are as defined

- <PresentationID> is the Presentation ID for the associated Presentation
- <PresResource> is the name of a Resource as referenced in Presentation/ResourceLibrary/Resource/LocalSource for that Resource. It must be unique within the associated directory (i.e., /Resource or a Presentation directory)
- <APID> is the APID for the associated Digital Asset.
- <DRM> is DRM Name as per [DSystem], Section 17.
- <MediaApplicationID> is the Media Application ID for the associated Media Application
- <AppResource> is the name of a Media Application Resource. It must be unique within the associated directory (i.e., /Resource or a Presentation directory)
- <AppName> is the name of the Application. It must correspond with Application/LocalSource in the Media Application Description document
- <Image URN> is the Image URN as defined in [DMeta] Section 4.3

- <image name> is a name for the image, including a file extension corresponding to the image type. <image name> must be unique within the /Images directory

4.3.2 SMPTE Media Package objects

4.3.2.1 Table of Contents

The Table of Contents Part specifies the list of presentations included within the SMPTE Media Package.

The `TableOfContents` element is the single root XML node of the Table of Contents Document, which is stored as a Part in the OPC/Zip container. The Table of Contents Part shall contain an XML document conformant with the XML Schema Definition file at the following location:

```
http://www.smpte-ra.org/schemas/2053/2011/MediaPackage/2053a-Media-Package-TableOfContents.xsd
```

A DMP SHALL contain a TableOfContents element.

The TableOfContents element SHALL be present in the DMP as follows:

Element	Attribute	Definition	Type	Card.
TableOfContents				
	Version	Version of the TableOfContents.		
	Source	DMPID for this DMP		
MediaApplications		Reference to Media Applications. Instances MAY be included.		0..n
PresentationRef		There SHALL be a PresentationRef instance for every Potential Media Presentation		1..n

A `PresentationRef` element contains the basic information about a presentation included in the SMPTE Media Package so that a consumer or device may select an appropriate presentation from the Table of Contents to download or play. It also contains basic metadata about that presentation.

There SHALL be a PresentationRef element for each Potential Media Presentation.

PresentationRef SHALL be populated as following

Element	Attribute	Definition	Type	Card.
PresentationRef				
	Id	PresentationID for the Media Presentation described by this element.		
	VersionRequired	The `VersionRequired` attribute specifies the version of the Presentation part that is expected. If this Table of Contents reference requires a higher version number than the stored Presentation document it references, then the newer version of the Presentation document should be downloaded to replace the older version.		
Title		Title of Media Presentation. This is for debugging and not for operational DMP use. Title SHOULD be equal to TitleDisplayUnlimited from one instance of LocalizedInfo in BasicMetadata.		
DescriptiveMetadata		An internal reference to MetadataMovie document (Container Required Metadata) for this presentation.		

4.3.2.2 Presentation Description

A Presentation Description Document as per [SMPTE2053] SHALL be present for each Potential Media Presentation in the DMP.

The Presentation Part shall contain an XML document conformant with the XML Schema Definition file at the following location:

```
http://www.smpte-ra.org/schemas/2053/2011/MediaPackage/2053c-Media-Package-PresentationRef.xsd
```

The Presentation element SHALL be populated as follows:

Element	Attribute	Definition	Type	Card.
Presentation				
	Version	This value SHALL equal TableOfContents/PresentationRef/@VersionRequired attribute for the PresentationRef associated with this Presentation.		
ResourceLibrary		Resources within the Presentation.		
DescriptiveMetadata		An internal reference to MetadataTail.		0..1
MediaApplications		Reference to Media Applications. Instances MAY be included.		0..n

Note that the DRM element is not required because licensing is handled through the DCC. The TrackGroup element is no required because track references are in MetadataMovie.

TableOfContents and Presentation are published together. If Version and VersionRequired do not match (e.g., after download), then the Table of Contents is not correct. This makes it possible to refer to [SMPTE2053] data with a single version value.

ResourceLibrary SHALL contain a Resource element for each Part of the DMP that is not one of the following

- Table of Contents document
- Presentation Description document
- Media Application Description document
- Container Mandatory Metadata
- Container Optional Metadata
- DCC
- Media Application

ResourceLibrary/Resource/@Id SHALL include the DECE identifier for the Resource, if any.

ResourceLibrary/Resource/@LocalSource SHALL be the filename of the Resource in accordance with Naming defined in Section 4.3.1.2.

ResourceLibrary/Resource/@Version SHALL be the Version of the Resource. Version SHALL increase with each revision.

Other elements and attributes of ResourceLibrary/Resource SHOULD NOT be included.

4.3.3 Media

An ODMP MAY contain DCC files.

There SHALL be exactly one DCC containing a video track associated with each Media Presentation.

4.3.4 Metadata and Images

Container Metadata as defined in [DMeta] contains information required to by Devices to select Presentations, select default tracks and provide other data-related functions (e.g., chapters). The principal element for Container Metadata is MetadataMovie. This is required in DCCs.

4.3.4.1 MetadataMovie

MetadataMovie in a DMP contains complete information about a Media Presentation. This includes where a track can be found (i.e., which DCC) and information needed to properly play the Media Presentation (e.g., chapters and default track selection information). Note that DCCs contain MetadataMovie, but since they do not necessarily describe all tracks information for playback might be incomplete. Therefore, it is generally necessary to use the metadata in the DMP.

A DMP SHALL contain MetadataMovie for all Presentations that are referenced by the Table of Contents.

The additional constraints apply to MetadataMovie within a DMP:

- The PresentationID attribute SHALL be present
- Within each instance of //MetadataMovie/TrackMetadata/Track/[Audio|Video|Subtitle],
    - TrackIdentifier/Namespace SHALL be 'DECE'
    - TrackIdentifier SHALL be the APID for the DCC containing that track
    - TrackReference SHALL be the track reference within the DCC associated with that APID

- When referring to an image in a Container,
    - //MetadataMovie/TrackMetadata/Track/Image/TrackReference SHALL be the APID for the DCC with that image
    - //MetadataMovie/TrackMetadata/Track/Image/TrackIdentifier SHALL be set with Namespace='DCCImageRef" and Identifier SHALL be the image reference as per [DMeta] 4.3.
- When referring to images in a DMP and not in a Container
    - //MetadataMovie/TrackMetadata/Track/Image/TrackReference SHALL be the DMPID for the DMP. Note this allows the image to be found, even when the metadata is separated from the DMP
    - //MetadataMovie/TrackMetadata/Track/Image/TrackIdentifier SHALL be set with Namespace='DMPImageRef", Identifier SHALL be the image reference as per [DMeta] 4.3, and Location SHALL be the filename of the Image within the DMP.

4.3.4.2 Storing Images

A DMP SHALL contain images in accordance with [DMeta] Container requirements. For the avoidance of doubt, [DMeta] 4.3 only applies to images within a Container and therefore does not apply to a DMP.

A DMP SHALL store images as files.

A DMP SHOULD use filenames for images corresponding with [DMeta] 4.3.

4.3.5 Base Locations and Licenses

Base Locations are required for licensing and for assistance with purchasing the Right associated with a DCC. DRM Licenses are required to play encrypted Content.

Note that this document describes where the format supports Base Locations and DRM Licenses. [DDevice] provides specific instructions on rules for storage. These rules can change over time.

4.3.5.1 Base Location in a DMP Part

The DMP SHALL have Base Location information for each Presentation in the DMP.

Base Location Parts SHALL be XML documents containing a BaseLocations element as defined in [DMeta], Section 4.

BaseLocations Part names SHALL be as defined in Section 4.3.1.2.

4.3.5.2 Licenses in a DMP Part

DRM License Parts are added as licenses are added to a Presentation. This can be done by the Device or the DSP.

DRM License Parts SHALL be binary files contain a single 'pssh' box as defined in DMedia.

DRM Licenses SHALL be organized by DRM in the /Licenses directory as defined in Section 4.3.1.2.

The License name SHALL correspond with the APID associated with that License.

4.3.6 Constraints on DCCs within DMPs

ZIP compression and encryption is not applied DCCs or the central directory. In ZIP parlance, compression method 0, "stored," is used. To be clear, this applies to DCCs in their entirety, not internal components of the DCC that are compressed and/or encrypted in accordance with [DMedia].

DCCs SHALL NOT be compressed ZIP objects.

DCCs SHALL NOT be encrypted ZIP objects.

DCCs SHALL be stored in Zip64 format.

4.3.7 Media Applications

The DMP MAY contain Media Applications as defined in [SMPTE2053].

END