Device Specification

Version 2.0 14 November 2014

Device Specification Version 2.0

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 28 December 2014.

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.

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

Device Specification Version 2.0

Contents

1 Document Description

1.1 Scope

This document specifies mandatory and optional features of DECE Devices; the features are operational when the Device joins a DECE Account via a domain-bound DRM Client.

This document also defines a “Common Player” and a “Common Streaming Player”. These definitions are suitable for reference by other ecosystems that play formats defined in [DMedia], [DStream], [DMeta] and so forth.

1.2 Document Organization

This document is organized as follows:

1. Introduction—Provides background, scope and conventions
2. DECE Devices and DECE Ecosystem – Describes how DECE Devices interact with other elements of the Ecosystem
3. Communications – Internet communications and browser support
4. Adding and Removing Devices from Account
5. Content Rights Purchase
6. Container Fulfillment – process for locating DECE Common File Format (CFF) Containers (DCC) and downloading them
7. DRM License Acquisition
8. Playing Content – Device requirements and limitations on decoding and presenting media
9. User-Related Requirement – Additional user interface functions
10. DLNA – Information on DECE Devices interacting with Digital Living Network Alliance (DLNA) devices
11. DECE Media Package (DMP) Support – Describes Device requirements related to DMPs
12. File Management – Information about placement and management of files

1.3 Document Notation and Conventions

Except where noted, notations and conventions are as per DECE Coordinator API Specification.

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 see that work.

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.

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 Normative References

1.4.1 DECE References

[DSystem]	System Specification
[DCoord]	Coordinator API Specification
[DMeta]	Content Metadata Specification
[DMedia]	Common File Format& Media Format Specification
[DSecMech]	Message Security Mechanisms Specification
[DDMP]	Media Package Specification
[DCManifest]	Common Media Manifest Metadata Specification

1.4.2 Other Normative References

[IANA-LANG]	IANA Language Subtag Registry. http://www.iana.org/assignments/language-subtag-registry
[RFC2141]	IETF RFC 2141, URN Syntax, May 1997. http://tools.ietf.org/html/rfc2141
[RFC2460]	IETF RFC 2460, Internet Protocol, Version 6 (IPv6) Specification, December 1998. http://tools.ietf.org/html/rfc2460
[RFC2616]	IETF RFC 2616, Hypertext Transfer Protocol -- HTTP/1.1, June 1999. http://tools.ietf.org/html/rfc2616
[RFC2617]	IEFT RFC 2617, HTTP Authentication: Basic and Digest Access Authentication, June 1999. http://tools.ietf.org/html/rfc2617
[RFC2782]	IETF RFC 2782, A DNR RR for specifying the location of services (DNS SRV), February 2000. http://tools.ietf.org/html/rfc2782
[RFC4346]	IETF RFC 4346, The Transport Layer Security (TLS) Protocol, Version 1.1, April 2006, http://tools.ietf.org/html/rfc4346
[RFC4647]	Philips, A., et al, *RFC 4647, Matching of Language Tags*, September 2006. http://www.ietf.org/rfc/rfc4647.txt
[RFC5646]	Philips, A, et al, *RFC 5646, Tags for Identifying Languages*, IETF, September, 2009. http://www.ietf.org/rfc/rfc5646.txt
[MPEG4S]	ISO/IEC 14496-1:2010, "Information technology — Coding of audio-visual objects — Part 1: Systems"
[UNICODE]	UNICODE 6.0.0, "The Unicode Standard Version 6.0", http://www.unicode.org/versions/Unicode6.0.0/

1.4.3 Informative References

[ISO-P2H]	ISO/IEC Directives, Part 2, Annex H: http://www.iec.ch/tiss/iec/Directives-Part2-Ed5.pdf
[UPNPCDS3]	*ContentDirectory:3 Service Template Version 1.01*, September 30, 2008, www.upnp.org/specs/av/UPnP-av-ContentDirectory-v3-Service.pdf

1.5 Terminology and Requirements Scope

Device-related terminology is defined in [DSystem].

DCCs may be contained in DECE Media Packages (DMPs). In all cases, except where noted, when a DCC is discussed this normatively refers to a DCC either by itself or as part of a DMP.

1.6 XML Change Management

DECE Devices SHALL comply with XML Change Management defined in [DSystem], Section 1.6.

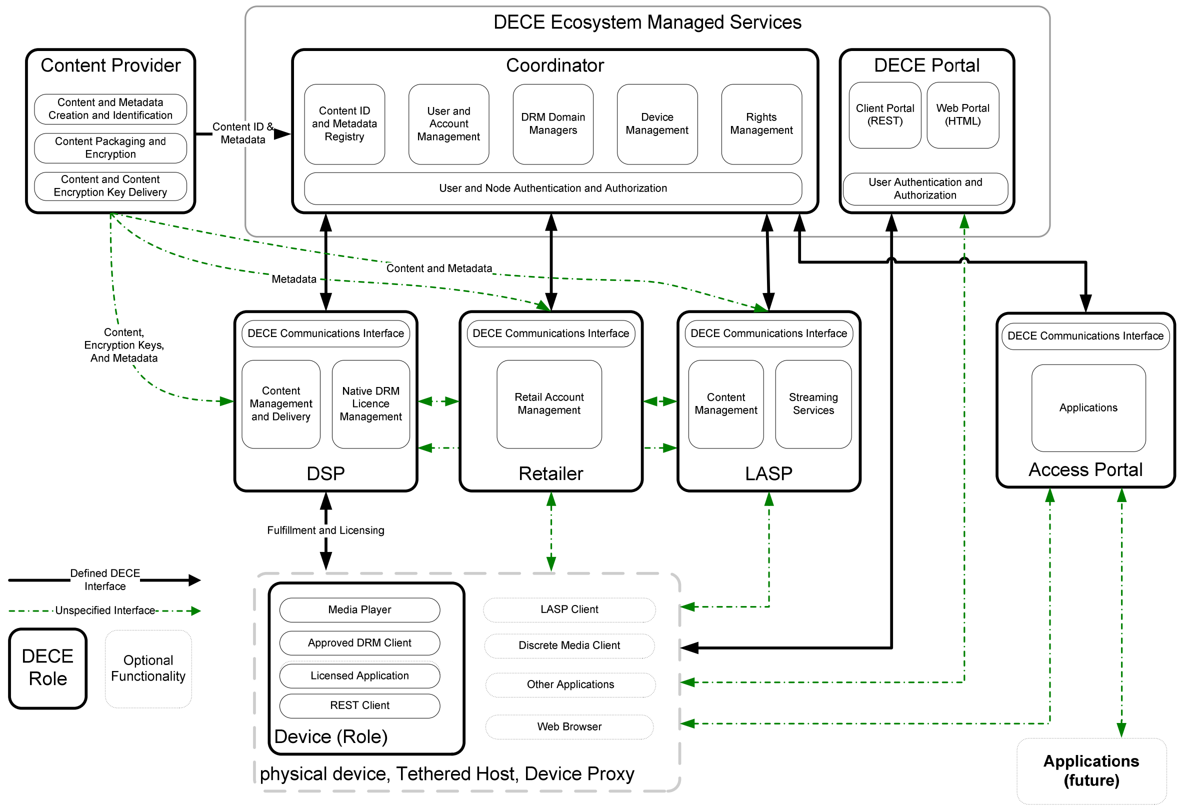
2 DECE Devices and DECE Ecosystem

This specification defines functionality associated with the “Device Role” which is specified in this document. A DECE Device is a hardware or software product or combination of products that implement a Device Role. DECE Devices include a DRM Client and a Licensed Application. DECE Devices are produced by Client Implementers.

As illustrated below, the DECE Device interacts with several components of the Ecosystem, such as

- Device Portal via REST APIs and/or Web Portal using a Browser
- DSPs to obtain content and licenses
- Coordinator for DRM domain management (e.g., joining the Ecosystem)

DECE Devices may, via non-DECE interfaces including Proxies, also have interfaces to Retailers and LASPs (for streaming).


DECE Ecosystem Managed Services
Content Provider
Content and Metadata Creation and Identification
Content Packaging and Encryption
Content and Content Encryption Key Delivery
Content ID & Metadata
Coordinator
Content ID and Metadata Registry
User and Account Management
DRM Domain Managers
Device Management
Rights Management
User and Node Authentication and Authorization
DECE Portal
Client Portal (REST)
Web Portal (HTML)
User Authentication and Authorization
Content and Metadata
Metadata
Content, Encryption Keys, And Metadata
DECE Communications Interface
Content Management and Delivery
Native DRM Licence Management
DSP
DECE Communications Interface
Retail Account Management
Retailer
DECE Communications Interface
Content Management
Streaming Services
LASP
DECE Communications Interface
Applications
Access Portal
Fulfillment and Licensing
Defined DECE Interface
Unspecified Interface
DECE Role
Optional Functionality
Media Player
Approved DRM Client
Licensed Application
REST Client
Device (Role)
LASP Client
Discrete Media Client
Other Applications
Web Browser
physical device, Tethered Host, Device Proxy
Applications (future)

The DECE Coordinator manages DECE Devices as part of Users' Accounts. It counts DECE Devices towards an Account's maximum allocation. A DECE Device with multiple DRM Clients would be managed by the Ecosystem as multiple DECE Devices. For example, a general purpose computer running three DRMs would count as three DECE Devices.

Separate from the DRM-specific interfaces, the DECE Device can communicate with the DECE Coordinator in three possible ways:

- To the Web Portal, using HTML and username/password authentication [DSecMech], Section 6;
- To the Device Portal, using the DECE Coordinator API [DCoord];
- Via an Access Portal using a proprietary Device-Retailer interface.

Which communication paths are required for various functions are described elsewhere in this specification.

When a Device joins a DECE Account, DECE records the unique identity of the DRM Client on that Device; to the DECE Coordinator, the identity of the Device is equivalent to the identity of the DRM Client on the Device. A physical device containing multiple DRM Clients either from different DRMs or from the same DRM but with different Native DRM Client IDs would be managed by the Ecosystem as if it were multiple Devices; the DECE Coordinator counts Devices towards an Account's maximum allocation.

DECE functionality may reside either within the DRM Client or in other DECE-aware applications, such as a Licensed Application (e.g., Media Player or Download Manager.)

The software in the DECE Device other than the DRM Client that performs functions specified by DECE is called a Licensed Application.

In any DECE Device implementation, DRM decryption and playback function must be performed in a single physical device. For avoidance of doubt, playback function of a DECE Device may include re-encryption of content without decoding by an Approved Output Technology. These physical devices may be connected to Tethered Hosts, typically a general purpose computer in the possession of a User, or to a Device Proxy, typically a server under the control of the Client Implementer.

Unless otherwise prohibited, any function assigned to a DECE Device MAY be implemented on a Tethered Host, a Device Proxy or a combination. Playback distribution is not allowed as per Section 8.6.

Some DRM Systems offer the ability for multiple applications to access a single instance of a DRM Client. In this case, a DECE Device could have multiple Licensed Applications.

When creating a Licensed Application resource in the Coordinator, it is necessary to include Device information.

The Coordinator may consolidate multiple Licensed Application/DRM Client pairs into a single Device resource if the DRM Client has the same DRM ID ([DSystem] Section 5.4.1) and is in the same DRM Domain.

3 Communications Requirements

3.1 Internet Communications

Connected Licensed Applications SHALL be able to communicate with the DECE Coordinator. Licensed Applications that communicate directly with the Coordinator SHALL

- Comply with [DCoord] for all APIs used by the Licensed Application
- Enable all required DRM Client interfaces and APIs, as specified in [DSystem], including license acquisition, domain join and leave operations, and the DRM-specific triggers for these operations.

In the case of Tethered DECE Devices, these communications functions will be on a Tethered Host device that is physically separate from the physical device containing the DRM Client. In the case of DECE Devices that use Device Proxies, these communications functions will be on a Proxy server device that is physically separate from the physical device containing the DRM Client.

In order to locate a preferred DECE Coordinator endpoint, a Device can do a DNS lookup for the SRV record. Licensed Applications SHOULD use SRV Records in the Coordinator and Portal DNS entries as specified in [DCoord], Section 3 and [RFC2782]. When a Licensed Application authenticates, it SHALL do so using one of the following mechanisms:

- o HTTP Basic Authentication as defined in [RFC2617] for subsequent communications with the Coordinator, or
- o Obtain a Security Token from the Coordinator using the `SecurityTokenExchange` API as defined in [DSecMech], Section 8.

When using Security Tokens, Licensed Applications SHALL handle Security Tokens in accordance with [DSecMech], Section 3.5.

DECE Devices communicating with a Device Proxy SHALL implement Confidentiality and Privacy Mechanisms as per [DSechMech], Section 3.2.

DECE Devices SHOULD support IPv6, as per [RFC2460].

DECE Devices SHALL authenticate themselves to the Coordinator via one of the profiles defined in [DCoord], Section 2.6.

DECE Devices SHALL include an Application Authorization Token in every Coordinator API invocation in accordance with [DCoord] Section 2.6.

4 Adding and Removing Devices to and from Account

The process of adding a DECE Device to a DECE Account involves both interaction with the Coordinator and a DRM-specific interaction with the Coordinator's Domain Manager. These are described in the [DSystem], Section 7.3. Coordinator APIs for Domain operations are found in [DCoord] Section 9.

4.1 Device Join

Device Join operations are assumed to be performed by a User who has a DECE Account.

4.1.1 Authentication and Obtaining a Join Trigger

Licensed Applications SHALL provide at least one of the following mechanisms for authenticating and obtaining a Join Trigger:

- Device Standalone Join – designed for DECE Devices with usable keyboards, network access and the ability to implement DECE REST APIs. Tethered DECE Devices use this method from a Tethered Host.
- Node Initiated Join (formerly Web Portal Initiated Join) – designed for Devices with limited data entry, particularly numeric digit entry
- Proxy Join – designed for DECE Devices that use Device Proxies.

Licensed Applications MAY also implement the following:

- Point of Sale Join – allows DECE Retailers to perform a partial Join of DECE Devices to an Account.

4.1.1.1 Device Standalone Join

In a Standalone Join, the Licensed Application first authenticates, and then obtains the DRM-specific Join Trigger using REST APIs through the DECE Portal using the REST Interface.

The following applies to DECE Devices implementing Device Standalone Join.

The Licensed Application SHALL perform the following operations:

- Authenticate the User if not done so already

- Perform a LicAppCreate() function as defined in [DCoord]. {LicAppID} is returned in the URL reference to the newly created resource. Then perform a LicAppJoinTriggerGet() call as defined in [DCoord].

If a Licensed Application determines a User does not have a DECE Account, the Licensed Applications SHALL inform the User that a DECE Account is required prior to a Join Operation.

4.1.1.2 Node Initiated Join (formerly Web Portal Initiated Join)

A Node Initiated Join begins with a User authenticating to an authorized Node, such as the DECE Portal, and initiating the process of adding a DECE Device. The Node provides the User with a numeric '*Device Authentication Code*'.

The following applies only to DECE Devices implementing Node Initiated Join.

A Licensed Application supporting Node Initiated Join SHALL:

- Provide a means for the User to initiate the transaction and enter the Device Authentication Code

- Obtain a Security Token from the Coordinator using the Device Authentication Code variant of the `SecurityTokenExchange` API as defined in [DSecMech], Section 8.

- Perform a LicAppCreate() function as defined in [DCoord]. {LicAppID} is returned in the URL reference to the newly created resource. Then perform a LicAppJoinTriggerGet() call as defined in [DCoord].

Licensed Applications SHALL accept numeric Device Authentication Codes up to DCOORD_DEVICE_JOIN_CODE_MAX_LENGTH numerals. DCOORD_DEVICE_JOIN_CODE_MAX_LENGTH is defined in [DCoord], Section 9 as part of DeviceAuthToken-type definition and in [DCoord], Section 23.

During entry Licensed Applications SHOULD display Device Authentication Codes in groups of three digits.

4.1.1.3 Proxy-based Join

Some Licensed Applications perform Domain Join operations with the participation of a Device Proxy which obtains a Domain Join Trigger. Details of this operation are described in the [DSystem].

The interface between the Licensed Application and Device Proxy are not specified by DECE, but SHALL result in a Device resource posted at the Coordinator, and a Domain Join Trigger for the appropriate DRM being delivered to the Licensed Application, equivalent to LicAppCreate() as defined in [DCoord].

The Device Proxy is required to perform the same operations as a Connected Device including authentication as defined in Section 4 above; and LicAppCreate() and LicAppJoinTriggerGet() as defined in [DCoord], Section 9.

If a Device Proxy determines a User does not have a DECE Account, the Licensed Applications SHALL inform the User that a DECE Account is required prior to a Join Operation.

4.1.1.4 Point of Sale (POS) Join

Point of Sale Join (POS Join) allows Retailers to add Devices to a User's Account, and allows a Device to Join a DRM Domain without the User entering additional data. POS Join is subject to constraints on the Retailer that are not specified here. Point of Sale Join requires that a User have a DECE Account. It is the responsibility of the Retailer to ensure that an appropriate DECE Account exists prior to attempting the POS Join process.

From the Licensed Application perspective, the POS Join is similar to a Node Initiated Join. The difference is that `DeviceHandle` generated from information internal to the Device is used in lieu of Portal-provided Domain Join Code.

POS Join requires a common secret[1], called a *DeviceUniqueString*, shared between the Retailer and the Device. It should not be practical for a third party to obtain or derive the DeviceUniqueString, for example by reading a bar code on outside of the box. For example, a string is generated by the Client Implementer and shared with the Retailer; and a code is put on the box that allows the Retailer to identify that string.

The Retailer posts the DeviceUniqueString to the Coordinator, creating a temporary record. At a later time, the Licensed Application uses the DeviceUniqueString as part of requesting the Join Trigger, and at that time, the Coordinator uses this information to match the Licensed Application to the temporary Retailer-created record and creates a Device record.

<DeviceUniqueString> need only be unique within the organization referenced by <OrgID>.

A Licensed Application supporting Point of Sale Join SHALL:

- Provide a means for the User to initiate the transaction
- Obtain a Security Token from the Coordinator using the Device Unique String variant of the `SecurityTokenExchange` API as defined in [DSecMech], Section 8.

[1] This is reasonably protected, but not necessarily on par with highly protected secrets such as DRM keys.

- Perform a LicAppCreate() function as defined in [DCoord]. {LicAppID} is returned in the URL reference to the newly created resource. Then perform a DeviceJoinTrigger() call as defined in [DCoord].

DeviceHandle is constructed as follows:

'DeviceString/'+<DeviceUniqueString>

<DeviceUniqueString> is Device Unique String defined in [DCoord], Section 9.4.3.4.

4.1.1.5 Superdistribution-based Join

This is not a distinct Join mechanism, but is a special case precursor to other Join operations.

The DECE Device receives a DCC before the Device is Joined to a DECE Domain. When the User attempts to play the DCC, the Licensed Application SHOULD offer the User the opportunity to Join the Device to a DECE Domain.

At this point, the Join becomes a Join by one of the other described mechanisms.

In the contingency that the DECE Device's User does not have a DECE Account, the Licensed Application SHOULD provide the User information on how to obtain a DECE Account.

4.1.2 DRM Join

4.1.2.1 DRM Join Operations

Licensed Applications SHALL be able to join a DRM Domain associated with a DECE Account, using the DRM's domain join mechanism.

Licensed Applications SHALL provide via DRM-specific mechanisms identification as follows:

- manufacturer and model, where manufacturer and model are sufficient to disambiguate Licensed Applications, otherwise
- manufacturer, model and Licensed Application identification.

Licensed Applications SHALL provide via DRM-specific mechanisms the `LicAppHandle`.

The application identifier is required when multiple applications could exist on a single device and must be distinguished.

Note that these data are not the `LicAppID` found in the `LicApp` resource.

4.1.3 Post DRM-Join Functions

If a DRM Join is unsuccessful, the Licensed Application SHALL remove residual data obtained as part of the Join process, including but not limited to Security Tokens.

4.1.4 Licensed Application Handle

A Device record in the Coordinator can have multiple Licensed Applications.

To limit access on certain functions, it is necessary to have a modestly protected piece of information shared between the Coordinator and the Licensed Application. This is handled via a value called a Licensed Application Handle (`LicAppHandle` attribute) in the Licensed Application record. `LicAppHandle` is a random number, sufficiently large to differentiate the Licensed Application from other Licensed Applications in the physical device.

The Licensed Application SHALL generate `LicAppHandle` value sufficiently random and large to avoid collision with other `LicAppHandle` values in a `LicApp` resource in a `Device` resource.

4.2 Device Leave

This section describes the mechanism for a DECE Device to leave a DECE Account's Domain in an orderly fashion, called a *Verified* Leave. That is, the Coordinator, including the Domain Manager, knows the DECE Device is not active, and the DRM Client on the Licensed Application removes credentials such that Containers licensed to the Domain no longer play.

Circumstances such as theft, damage or loss may result in a DECE Device no longer being part of the DECE Account's, although Verified Leave process has not occurred. This is called an *Unverified Leave.* Unverified Leave does not have DECE Device involvement and is therefore not covered in this specification. Further details can be found in [DSystem], Section 7.3.4.

4.2.1 Leave Warning

Prior to removing a Device from a DECE Account, the Licensed Application SHALL provide a warning to the User. This warning SHALL contain at least the following information:

- Content licensed for that DECE Device's Domain will no longer play

Note that a Device Move is a special case of Leave, so this notice is also part of a Move.

4.2.2 Obtaining a Leave Trigger

DRMs that allow or require a Leave Trigger to leave a DECE Domain can obtain a Leave Trigger.

Licensed Applications MAY provide at least one mechanism for obtaining a Leave Trigger.

The means of obtaining a Leave Trigger are as follows:

- Device Standalone Leave
- Proxy Leave

4.2.2.1 Device Standalone Leave

In a Standalone Leave, the Licensed Application directly obtains the DRM-specific Leave Trigger using REST APIs through the DECE Portal using the REST Interface.

The following applies to Licensed Applications implementing Device Standalone Leave:

- When obtaining a Leave Trigger, the Licensed Application SHALL perform a LicAppLeaveTriggerGet() function as defined in [DCoord], Section 9.

4.2.2.2 Proxy Leave

Some Licensed Applications perform Domain Leave operations with the participation of a Device Proxy which obtains a Domain Leave Trigger. Details of this operation are described in [DSystem] Section 7.3.

The interface between the Licensed Application and Device Proxy are not specified by DECE, but SHALL result in obtaining a Domain Leave Trigger for the DRM Client, equivalent to LicAppLeaveTriggerGet () as defined in [DCoord]. Note the Device Proxy must perform the LicAppLeaveTriggerGet (), but Device Proxy specification is outside the scope of this spec.

4.2.3 DRM Leave

Licensed Applications SHALL be able to leave a DRM Domain associated with a DECE Account, using the DRM’s domain leave mechanism.

Licensed Applications SHALL perform a DRM-specific Leave.

4.2.4 Device Leave Cleanup

When a DECE Device leaves a DECE Domain, the Licensed Application SHALL remove the following:

- Account-specific, Domain-specific and User-specific identification information. This includes removing DECE Security Tokens in accordance with [DSecMech], Section 3.5, and all data unique to the Account that facilitates playing DCCs.

After Domain Leave, DCCs licensed to the Account Domain SHALL be unplayable.

4.3 Device Move

Device Move is a combination of a Device Leave and a Device Join.

Device Move is generally initiated by an attempt to Join a DECE Device to another Account.

A Licensed Application SHALL perform a complete Device Leave prior to performing a Device Join.

4.4 Multiple Licensed Applications and DRM Clients

Some Licensed Applications are capable of accessing multiple DRM Clients. Some DRM Clients support the use of multiple Licensed Applications.

A Licensed Application that uses multiple DRM Clients SHALL perform a DRM Join for each DRM Client.

A Licensed Application that uses multiple DRM Clients SHALL perform a DRM Join in only one DECE Domain.

A Licensed Application SHALL perform a Leave operation on all associated DRM Clients before Joining the new Domain.

DRM Clients SHOULD prevent multiple instances of the DRM Client being in separate DECE Domains on a single hardware device.

A Licensed Application SHOULD NOT allow multiple DRM Clients to be in different DECE Domains on a single hardware device.

DRM Clients SHALL enable any mechanisms available that prevent or can help prevent multiple instances or multiple applications of the DRM to join independent DECE Domains on a piece of physical hardware. For example, DRM systems that can provide a unique ID that is mapped to the physical hardware must enable such mechanisms.

Any Licensed Application MAY perform a LicAppCreate().

5 Content Rights Purchase Support

The process of obtaining content Rights (i.e., purchasing) is not part of this specification as the device has no normative role in the process, with one exception. That exception relates to superdistributed content and is described below.

5.1 Purchase of Content Rights

Content Rights are sold by DECE Retailers and posted to the Coordinator. In general, any involvement of a DECE Device in the purchase process is outside of the scope of DECE specification. Interfaces are considered proprietary to the Retailer and purchase applications. This section assumes a purchase application associated with a DECE Device running on the same physical device or otherwise implemented in conjunction with the Licensed Application. The purchase application may provide information to a Licensed Application.

A Retailer may return information to a purchase application that can help the Licensed Application download the DCCs associated with the purchased Right. This is desirable because it saves the step of the Licensed Application locating the DCC (see DCC Acquisition below). For example, the information returned may include one or more of the following:

- An HTML page containing links leading to DCC download,
- An HTML page containing a link to a Download Manifest,
- A Download Manifest.

If the Licensed Application receives a Download Manifest, it is expected that a Download Manager on the Device is able to parse that document and proceed to download the files. The format of the DECE Download Manifest is defined in DECE System Design [DSystem].

If a purchase application associated with a Device attempts to purchase Rights before the Device has joined any DECE Account, the application may give the user the opportunity to join the Device to a DECE Account. This process is also outside the scope of this specification.

5.2 Purchasing Rights for Superdistributed Content

DCCs can arrive at DECE Devices through Superdistribution (see [DSystem], Sections 1.4 and 15), possibly as part of a DMP. Typically, a User is expected to obtain a DCC and attempt to play it on one of their DECE Devices. As the Superdistributed file does not contain a license for the User's Account and the Device's DRM, it will not play. This process is described under DRM License Acquisition below.

If the User wishes to purchase a Right to play the DCC, it is necessary to identify a Retailer that sells Rights to the Superdistributed DCC. Although a general mechanism for locating a Retailer who sells the Rights to a DCC is not specified by DECE, it is possible to find one such Retailer by using the a Purchase URL (PURL) that can be derived from information in the DCC. Note that DMPs can contain PURL information. This can assist purchasing.

5.2.1 Purchase URL (PURL) Construction

The Digital Asset may optionally include a Base PURL Location that can be used to create a PURL.

The Purchase URL provides a location where a Right may be purchased via a web browser. There is no implicit guarantee that the Right can be purchased (e.g., Retailer may have stopped selling that content), but there is a guarantee that if the Right is purchased, the DCC with the PURL will be licensable under that Right.

Base PURL Location is read or written (accessed) in accordance with the following:

- From a Multi-track DCC that is not in a DMP, the Licensed Application SHALL access Base PURL Location from the 'bloc' Box of DCC as defined in [DMedia], Annex E.
- From a DMP, the Licensed Application SHALL access Base PURL Location, in the BasePurlLocation element of the BaseLocations Part of the DMP as defined in [DDMP], Section 4.3.5.

If the Digital Asset includes a Base PURL Location, a Licensed Application MAY construct the PURL in accordance with [DSystem], Section 8.3.3 and use a web browser to enable purchase.

At least once, a Licensed Application SHALL obtain <decedomain> from the Coordinator using DeviceDecedomain().

The Licensed Application SHALL validate that Base PURL Location uses RFC-conformant syntax and ensure the domain name ends with <decedomain> as per [DSystem], Section 8.3.3. This prevents situations where the BasePurlLocation would result in a PURL that refers to a domain not under DECE control. For example, a BasePurlLocation of 'phishingdomain.com/' would be rejected as it would result in a PURL such as 'http://purchase.phishingdomain.com/.uvvu.com/?index.html?apid=…'

5.2.2 Alternate Mechanisms for locating Retailers

Although not specified by DECE, a Licensed Application may use other methods to locate a Retailer, including use of third party services, or having a pre-existing relationship with one or more DECE Retailers.

5.2.3 Base Location Updates

The following applies only to Devices that are Joined to an Account.

After purchase, the Licensed Application SHALL query the Rights Token to see if `LicenseAcqBaseLoc` in the Rights Token is different from `BaseLocation` field in the Digital Asset as defined in [DMedia], Annex E.

Base Location is read or written (accessed) in accordance with the following:

- From a Multi-track DCC that is not in a DMP, the Licensed Application SHALL access Base Location in the 'bloc' Box of DCC as defined in [DMedia], Annex E.
- From a DMP, the Licensed Application SHALL access Base Location, in the BaseLocation element of the BaseLocations Part of the DMP as defined in [DDMP], Section 4.3.5.

If the `LicenseAcqBaseLoc` obtained from the Rights Token is different from the Digital Asset's Base Location, Licensed Applications on devices that support Outbound File Transfer replace the Digital Asset's Base Location with `LicenseAcqBaseLoc`. That is, 'bloc' is updated in DCCs and a BaseLocation element is updated in DMPs.

Licensed Applications SHALL replace the BaseLocations Part with a new Part with the same name, if necessary. Note that on Devices that do not support Outbound File Transfer the BaseLocations Part is not necessarily in the DMP.

In a DMP, Base Locations can be updated through the DMP Part update mechanism.

Licensed Application on devices that do not support Outbound File Transfer SHALL use the new Base Location, although they do not need to write it to the DCC.

This is necessary because the Base Location is used for licensing and an incorrect Base Location will cause unnecessary redirects as part of the licensing process.

5.2.4 License Acquisition after Download

The following applies only to Devices that are Joined to an Account.

After purchase, a Licensed Application SHALL attempt to license the DCC that is downloaded. See License Acquisition, below, for requirements associated with license acquisition after download.

6 Digital Asset Fulfillment

DECE supports several methods of delivering content to Devices and incorporating that content into the Device's storage. Fulfillment is the term used to describe the process of delivering licensed DECE Content in the form of DCCs and DMPs to the Device.

DCCs and DMPs (including DMP Parts), are collectively referred to as Digital Assets.

Devices SHALL be able to acquire any DCCs consistent with their supported profiles from a DSP.

Devices SHALL be able to acquire any DMPs consistent with their supported profiles from a DSP. Note that Devices that do not support Outbound Transfer are not required to maintain information in DMP format.

In this context, the term 'acquire' is used to indicate that the Device can gain access to the Digital Asset in some manner. For example, it can download the Digital Asset, it can access the Digital Asset over a LAN, it can be side-loaded with a storage device with the Digital Asset, or it can otherwise gain access to that Digital Asset.

6.1 Initiating Fulfillment and Streaming

Fulfillment and Streaming can be initiated through a Retailer, through the Web Portal or via a Rights Locker query to the Device Portal. The Retailer and Web Portal cases are web-based or use proprietary interfaces between the Retailer and the DECE Device; and are outside the scope of this specification (see [DSystem], Section 11.)

Before initiating a download or stream, a Licensed Application must first obtain either a URL pointing to a download or stream web site (called a Fulfillment Web Location) or a URL point to a manifest file that includes information for downloading or streaming one or more Digital Assets.

These locations can be obtained from the Coordinator via the Rights Token query APIs. Licensed Applications MAY support RightsTokenGet() as defined on [DCoord], Section 7.

The particular relevant elements of the Rights Token are `FulfillmentWebLoc` and the `FulfillmentManifestLoc`. At least one of each will exist, and there may be more than one. These location elements each contain a URL and optionally an element called Preference defined as an integer. Preference defines an ordering.

Licensed Applications SHOULD use the URLs with the following precedence:

1. URLs with lower `Preference` value are used before URLs with higher `Preference` value

2. URLs with `Preference` are used before URLs without Preference

3. Two or more URLs with the same `Preference` may be used in any order

4. Two or more URLs without `Preference` may be used in any order

`FulfillmentWebLoc` MAY be passed to a browser in the Licensed Application.

`FulfillmentWebLoc` MAY be passed outside of the Licensed Application. For example, it may be passed to another device with a web browser.

6.2 Download Manager, Web Download, and Streaming

6.2.1 Protocol

Protocol applies to both Download Manager, Web Download and Streaming

Licensed Applications that support Download Manager SHALL support HTTP and HTTPS in accordance with [RFC2616] and TSL 1.1 [RFC4346].

Licensed Applications that support Common Streaming SHALL support HTTP and HTTPS in accordance with [RFC2616] and TSL 1.1 [RFC4346].

Licensed Applications SHOULD support Range GETs for resuming partial downloads [RFC 2616], Section 14.35 ‘Range’.

6.2.2 Download Manager and Stream Manager

Download Manager is used to download Content. Stream Manager is used to stream Content using Common Streaming.

The Download Manager knows which files to download based on a Fulfillment Manifest and Fulfillment Manifest File as defined in the System Design Specification [DSystem] Section 11.1.

The Stream Manager knows which files to stream based on Fulfillment Manifest and Fulfillment Manifest File as defined in the System Design Specification [DSystem] Section 11.1.

The first step is to obtain the Fulfillment Manifest File. It is obtained using HTTP GET as specified under Protocol above.

The DCC download process is at the discretion of the Licensed Application.

The DMP download process is at the discretion of the Licensed Application, although the end product of a downloaded SHALL be complaint with [DDMP], except for Devices that do not support Outbound File Transfer.

A Licensed Application MAY interact with the User to select which files to download or stream.

Licensed Applications SHOULD support continuation of downloads that were interrupted.

A Fulfillment Manifest is only valid for the download or stream session; that is, for one or more downloads in parallel or sequentially, or streaming, without limited interruption. Note that downloading a DMP including its forced download components is considered a single session. It is important that manifests be current as old manifests could cause incorrect files to be downloaded, including the wrong Content.

A DECE Device SHOULD obtain a Fulfillment Manifest each time it begins a new download operation.

A DECE Device SHOULD obtain a new Fulfillment Manifest when it encounters indications of a stale manifest such as inoperable links.

6.2.3 Web Download

Web download is via standard web download mechanisms.

To enable web download, fully functional (populated) DMPs must be provided by the DSP. Alternatively, a download manager must be used.

6.3 Digital Asset Download Options

Licensed Applications SHALL support Digital Asset acquisition from DSPs by either downloading directly from the DSP or by supporting the ability to transfer Digital Asset from devices that download directly from DSPs.

Licensed Applications SHOULD support Digital Asset acquisition via Superdistribution.

Licensed Applications MAY support Digital Asset acquisition via other mechanisms.

Licensed Applications SHALL support DMP acquisition from DSPs either by downloading directly from the DSP or by supporting the ability to transfer DMPs from devices that download directly from DSPs.

6.3.1 Download from DSP

Download is performed through a connection between the DECE Device and a DSP. DECE Devices include Tethered DECE Devices, although the connection may be performed by the Tethered Host.

A Connected DECE Device MAY support Direct Download of Digital Assets, either via Web Download or Download Manager, or both.

A DECE Device that supports download SHOULD support the Download Manager mechanism.

6.3.2 Separate Download and Copy

Download may be initiated by a physical device other than the DECE Device. The downloaded file is then copied to the DECE Device.

Retailers and DSPs may present mechanisms to download files to a User. For example, the Retailer may implement a web site with links to locations where DCCs may be downloaded. Alternatively, Retailers or 3rd parties might supply download applications that will download Digital Assets.

These mechanisms result in a Digital Asset available to a DECE Device.

DECE Devices SHOULD accept files downloaded via indirect downloads and copied to the DECE Device.

6.3.3 Other Loading Mechanisms

Tethered DECE Devices SHALL accept Digital Asset via a Tethered Host. DECE Devices MAY accept Digital Assets via copying. Copying is the process of delivering content to a device through a mechanism other than the Internet or tethering. Copying may occur via portable media or local wired or wireless connection. Sometimes the term sideloading is used in reference to copying to a device and should be interpreted the same as copying.

6.4 Progressive Download

Licensed Applications MAY begin playback during download.

6.5 License Acquisition after Download

After download, if a DCC is not already licensed for the DRM Client supported by the Device, the Licensed Application SHALL attempt to license that DCC. See License Acquisition, below, for requirements associated with license acquisition after download. In the case of a DCC within a DMP, this requirement refers only to DCCs meeting the playability conditions of Section 8.2.

Note that the requirements above apply to DCCs within DMPs, regardless of whether the DCC was downloaded as part of the DMP or separately.

6.6 DMP Download Requirements

Licensed Applications that download DMPs SHALL comply with the conditions of this section. That is, any Licensed Application that downloads DMPs must do so in the manner defined in this section. In particular, this section does not apply to Licensed Applications that only obtain Digital Assets via Other Loading Mechanisms, as per Section 6.3.3.

Licensed Applications that download DMPs that do not support Outbound File Transfer NEED NOT store files in a DMP. Note that the ODMP download is still required, but DMP Part downloads can be stored at the discretion of the Device.

DMP Download and Update processes and data structures are defined in [DSystem]. Licensed Applications that download DMPs SHALL comply with processes defined in [DSystem], Section 11.1.4, except where noted in this document.

6.6.1 Initial Download

Licensed Applications that download DMPs SHALL be capable of downloading the DMP as referenced in the Manifest.

Subsequent to downloading an ODMP, Licensed Applications that download DMPs SHALL identify all Parts Requiring Download and Download those items. Licensed Applications NEED NOT download Parts if the exact Parts are already in the DMP. For the avoidance of doubt, if a Part happens to be identical to another, but is used in a different place or manner it is not considered the same Part.

A Licensed Application that downloads DMPs MAY download any Resource.

It is not anticipated that the download process will corrupt Parts, however it is possible the wrong Part could be downloaded or some other problem results in a corrupted file. It is RECOMMENDED that Licensed Applications that download DMPs perform some validation on downloaded Parts. Other forms of verification are acceptable. For example, in lieu of hashing a large DCC, a Licensed Application might verify that the APID is correct.

Licensed Applications that download DMPs SHALL attempt to download all Items Requiring Download in one session. For avoidance of doubt, this indicates Devices are expected to download all components in parallel or sequence without delay.

6.6.2 Subsequent Download

Licensed Applications that download DMPs SHALL support DCC Subsequent Download.

Licensed Applications that download DMPs SHALL support Media Application Subsequent Download. Note that DECE does not currently support any Media Application technology.

6.6.3 DMP Update

Licensed Applications that download DMPs SHALL perform necessary DMP updates as indicated by a FulfillmentManifest document prior to or in conjunction with DCC and Media Application Subsequent Download. Update consists of downloading and saving in the DMP any DMP Parts that require updating as indicated by the Version mechanisms as defined in [DDMP] Section 4.2 and [DSystem], Section 11. This includes items described in FulfillmentManifest/DMPUpdateItems. Note that no updates are required when a FulfillmentManifest is downloaded as part of initial DMP Download.

Licensed Applications that download DMPs SHALL support DMP information update, defined in Manifest/DMPUpdateItems.

Licensed Applications that download DMPs MAY support DMP information update separate from Subsequent Download. For example, if a Device becomes aware that a new track is available, that Device might wish to update the DMP to reflect this.

6.6.4 DMP File Maintenance

When files are replaced, they can waste space in the DMP. 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 DMP (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. Information about what constitutes obsolescence of each Resource type is provided in [DSystem], Section 11.1.3.

Licensed Applications that download DMPs SHOULD remove Obsolete Parts from the DMP Obsolete.

7 DRM License Acquisition

7.1 Acquisition of Content License

Devices must be able to acquire a DRM license for any encrypted Content the Device wishes to play, regardless of which Retailer the content was originally purchased from, which DSP a DCC was originally downloaded from, or which LASP is streaming via Common Streaming.

To obtain a license in this circumstance, the Device locates a DECE DSP with a DRM License Server from which it can request and obtain a DRM-specific license for the DCC in question; such a DSP must (a) support the same DRM that the DECE Device supports, and (b) have rights to create licenses for the content in the DCC in question. There are two mechanisms for locating a license server and the DECE Device SHALL support both:

1. Base Location: using Base Location information in the DCC or DMP, or retrieved via streaming information in the Fulfillment Manifest

2. Coordinator-based referral: using information obtained from the Coordinator

The Device SHOULD first attempt to obtain a license using the first mechanism (Base Location), and only use the second mechanism (Coordinator-based location) if the first mechanism fails. Note that Common Streaming Devices only use Base Location.

Licensing requirements are the same whether a DCC is within a DMP or separate.

Licenses are stored in DMPs in their own Part as defined in [DDMP], Section 4.3.5. The same licenses apply for the DMP as apply for a multi-track DCC.

When performing Licensing functions for a Presentation in a DMP, the Licensed Application SHALL use the DCC that contains the Primary Video Track as per Section 11.1.2.

Note that future version may have provisions for licensing individual tracks.

7.2 License Acquisition Flow

This section defines the sequence of events associated with locating a license server and acquiring a license. An explanation of each step is provided below.

7.2.1 Support for License Acquisition Flow

There are three conditions that potentially require a licensing attempt by a DECE Device: Purchase, Ingest and Play.

Purchases performed by the Device using the PURL mechanism may result in a licensing attempt as per Section 5.2.3, Base Location Updates.

Ingest occurs when a DECE Device obtains a DCC by download, file copy, transfer through a tether or other transfer operation that results in a new DCC on that DECE Device. The goal of licensing upon ingestion is to increase the likelihood that a DCC is playable, even if the DECE Device is offline when a play is attempted (e.g., on an airplane without broadband). DCCs installed in a DECE Device prior to delivery to a User (i.e., Preloaded Content) are not considered 'ingested' in the context of this definition.

Play occurs when there is an attempt to play Content.

A DECE Device SHALL be Joined to a DECE Domain prior to attempting to acquire a license. Device Joining is described in Section 4.1, Device Join.

A DECE Device MAY attempt to license a file using *General License Acquisition Flow* at any time.

A DECE Devices SHALL comply with *General License Acquisition Flow* when a DCC is ingested into the DECE Device. This does not apply to preloaded content as per DECE System Design [DSystem] Section 15.

A DECE Devices SHALL only obtain licenses using information provided in Base Locations.

A DECE Device SHALL comply with the *General License Acquisition Flow* when attempting to play Content.

7.2.2 General License Acquisition Flow

The following flow chart defines the sequence of events associated with locating a license server and acquiring a license; this sequence is called the "General License Acquisition Flow". An explanation of each step is provided below.

The following conditions are assumed to hold before the beginning of the Flow:

- A DCC is present in the Device;
- The Device is joined to a DECE Account; and
- The Right to the Asset is present in the Coordinator, for the Account in question.

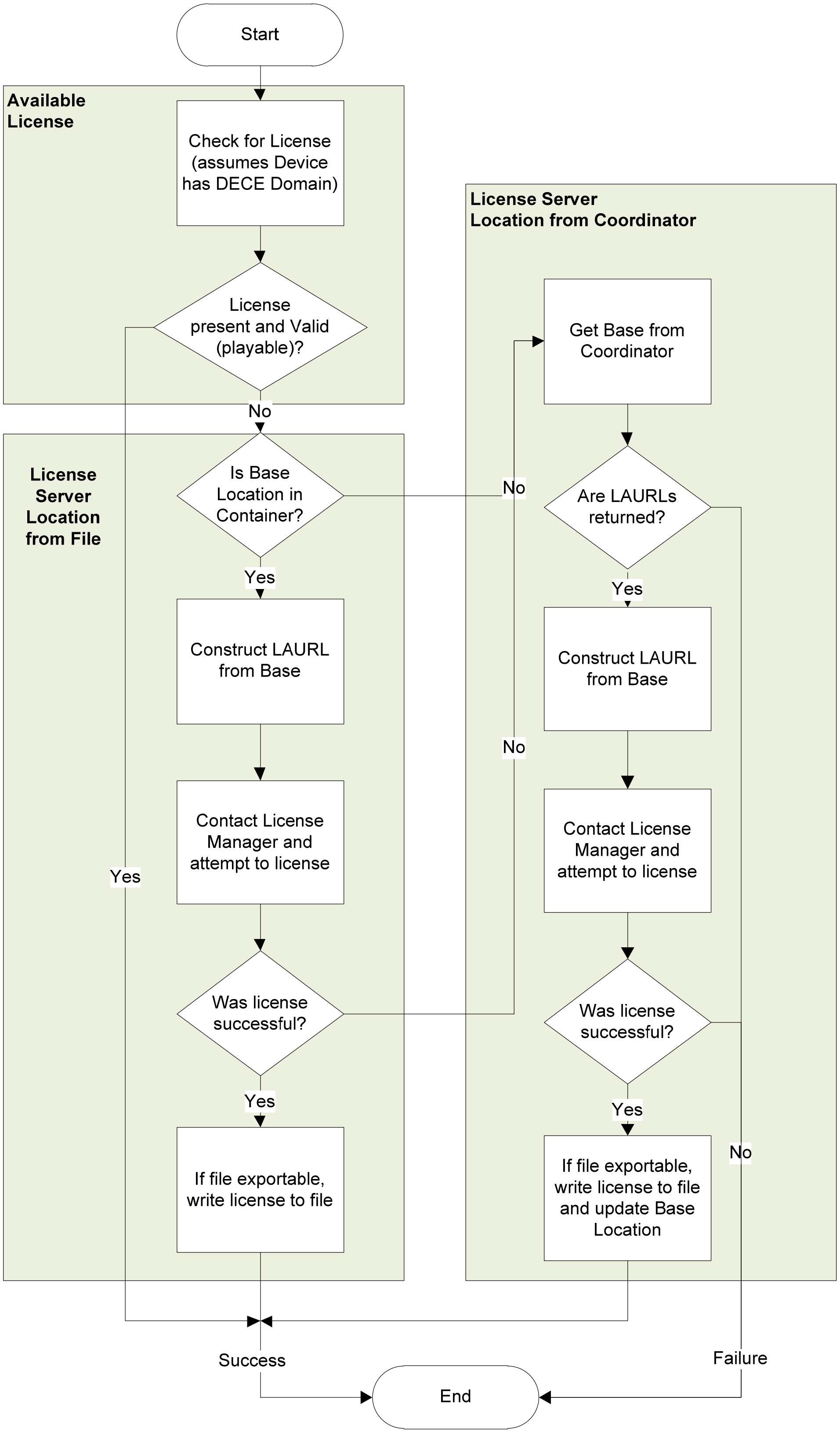
This flow is initiated at 'Start' when a DCC is ingested into a DECE Device, when there is an attempt to play or stream Content, or at any time the DECE Device otherwise determines a licensing operation is appropriate.

The first operation checks to see if a license is present. If so, the process is complete.

If not, it attempts to obtain a license using the Base Location to construct a LAURL and use that LAURL to locate a license server, and then obtain a license. If that operation is successful, the process is complete.

If license is not either initially available or available through the LAURL process, at attempt is made to locate the license server through the Coordinator and obtain the license at the indicated location.

If the attempt to obtain a license through the Coordinator fails, the overall operation fails and a license is not obtainable. Following failure, the DECE Device has the option of initiating a purchase operation as described above in Section 5, Content Rights Purchase.


Start
Available License
Check for License (assumes Device has DECE Domain)
License present and Valid (playable)?
No
Yes
License Server Location from File
Is Base Location in Container?
Yes
No
Construct LAURL from Base
Contact License Manager and attempt to license
Was license successful?
Yes
No
If file exportable, write license to file
License Server Location from Coordinator
Get Base from Coordinator
Are LAURLs returned?
Yes
Construct LAURL from Base
Contact License Manager and attempt to license
Was license successful?
Yes
No
If file exportable, write license to file and update Base Location
Success
Failure
End

7.2.3 License Server Location Obtained from DCC

A DECE Device SHALL be able to obtain Base Location information from a DCC, as defined in [DMedia], Annex Eand [DSystem], Section 8.3.

A DECE Device SHALL be able to obtain Base Location information from a DMP, as defined in [DDMP], Section 4.3.5 and [DSystem], Section 8.3.

A DECE Device that supports Common Streaming SHALL be able to obtain Base Location information from a Fulfillment Manifest, as defined in [DSystem], Section 11.1.4. Note that information is in FulfillmentManifest/StreamingInfo/BaseLocations.

License Server location information can be derived from the Base Location. If the Base Location information is present in the DCC, the Device SHALL be able to retrieve and act upon such information to request and obtain the License from the License Server.

The following steps are involved in locating a license server,

(1) the DECE Device retrieves the location information from the DCC,

(2) the DRM Client contacts the DRM-specific License Server with information is necessary for Rights verification.

(3) If the Domain has the Right to play the Content, a DRM-specific License is delivered.

7.2.3.1 License Acquisition Location (LALOC)

If a file needs to be licensed, the Base Location is identified

Base Location is read in accordance with the following:

- From a Multi-track DCC that is not in a DMP, the DECE Device SHALL read Base Location from the 'bloc' Box of DCC as defined in [DMedia], Annex E.
- From a DMP, the DECE Device SHALL read Base Location from the BaseLocation element of the BaseLocations Part of the DMP as defined in [DDMP], Section 4.3.5.
- For Common Streaming, a DECE Device that supports Common Streaming SHALL read Base Locations from FulfillmentManifest/StreamingInfo/BaseLocations

Assuming a Base Location, the License Acquisition Location (LALOC) is constructed. The LALOC is constructed from the Base Location as follows:

- License Acquisition Location (LALOC) SHALL be constructed as defined in [DSystem], Section 12.2.

- The DECE Device SHALL validate that LALOC uses RFC-conformant syntax and ensure the domain name ends with <decedomain> as per [DSystem] Section 8.3.4. This prevents situations where the BaseLocation would result in a LALOC that refers to a domain not under DECE control. For example, a BaseLocation of 'phishingdomain.com/' would be rejected as it would result in a LALOC of the form 'http://<DRMname>_license.phishingdomain.com/.uvvu.com.

7.2.3.2 Licensing

A DECE Device SHALL contact a DRM-specific license manager at the location specified by the LALOC and obtain a license using DRM-specific protocol.

If licensing succeeds, the DECE Device proceeds with conditionally writing the License as defined below.

If the licensing fails, the DECE Device proceeds as per Section 7.2.4 *License Server Location Obtained from Coordinator*.

7.2.3.3 Writing License

When streaming via Common Streaming, the DECE Device SHALL NOT write the license to any persistent storage. The following requirements apply only to download.

When a license is obtained by a DECE Device capable of Outbound File Transfer, when licensing a Presentation within a DMP the DECE Device SHALL write the license as a DMP Part as defined in [DDMP], Section 4.3.5.

When a license is obtained by a DECE Device capable of Outbound File Transfer when licensing Content not within a DMP, the DECE Device SHALL write the license as defined in Section 7.2.5, *License Management in DCC*.

The following definitions are used for requirements in this section:

- Total License Space is defined as the sum of all 'pssh' Boxes sizes and the 'free' Box in the 'moov' Box.
- License Space Consumed is the sum of 'pssh' Boxes sizes.
- Excess License Space is defined as the magnitude of the difference between License Space Consumed after and License Space Consumed before the license was added. For example, if 25KB was used before and 30KB were used after, the Excess License Space is calculated as 5KB = 30KB – 25KB.
- License Default Allocation is 20KB, for the 'pssh' Box in the 'moov' Box, unless otherwise specified in the [DSystem], Appendix B.

When writing a license to a DCC would result in expanding License Space Consumed and would result in exceeding the DECE Device's DRM's License Default Allocation, the DECE Device SHALL reformat the DCC to expand the Total License Space by at least the Excess License Space.

DECE Devices SHOULD have the ability to reformat DCCs to expand the Total License Space if there is not enough Available License Space to write a license.

7.2.4 License Server Location Obtained from the Coordinator

If Base Location is either not available, or does not lead to successful license acquisition, the Coordinator can provide a set of LALOCs for the asset, assuming that the DRM Client's Domain is part of a DECE Account that holds a Right for that DCC.

Use of LALOC is described in [DSystem] Section 12.2.2.

7.2.4.1 License Acquisition Location (LALOC)

If the DCC does not have a suitable License Server location, the DECE Device SHALL obtain locations from the either the Device Portal or an Access Portal. Access Portal interface to the Device is not specified by DECE.

DECE Devices obtaining License Server location information from the Device Portal SHALL use Rights Token GET APIs [such as, RightsTokenGet() and RightsTokenDataGet()] as defined in [DCoord], Section 7.

If Rights Token GET APIs fails the licensing operation has failed and the User should be informed and may be offered the opportunity to purchase the content as per Purchasing Content above.

The following assumes Rights Token GET APIs succeeds.

The particular relevant element of the Rights Token is `LicenseAcqBaseLoc`. LALOC is constructed from `LicenseAcqBaseLoc` as described above.

7.2.4.2 Licensing

A DECE Device SHALL contact a DRM-specific license manager at the location specified by the LALOC and obtain a license using DRM-specific protocol.

7.2.4.3 Writing License and Base Location

When streaming via Common Streaming, the DECE Device SHALL NOT write the Base Location to any persistent storage. The following requirements apply only to download.

When a license is obtained by a DECE Device capable of Outbound File Transfer, when licensing a Presentation within a DMP the DECE Device SHALL write the license as a DMP Part as defined in [DDMP], Section 4.3.5.

When a license is obtained by a DECE Device capable of Outbound File Transfer, when licensing Content not within a DMP, the DECE Device SHALL write the license as defined in Section 7.2.5, *License Management in DCC.*

When a license is obtained for a Presentation within a DMP by a DECE Device capable of Outbound File Transfer using a License Server Location obtained from the Coordinator, the DECE Device SHALL write the `LicenseAcqBaseLoc` obtained from the Rights Token into BaseLocations Part of the DMP as defined in [DDMP], Section 4.5.3.

When a license is obtained for a DCC not part of a DMP by a DECE Device capable of Outbound File Transfer using a License Server Location obtained from the Coordinator, the DECE Device SHALL write the `LicenseAcqBaseLoc` obtained from the Rights Token into `BaseLocation` field in the DCC as defined in [DMedia], Annex E.

7.2.5 License Management in DCC

When a license is to be written to a DCC or removed from a DCC, the DECE Device SHALL do so as follows.

7.2.5.1 Scheme

The section applies to Scheme-signaled DRM-specific information.

Within a DCC, licenses are in ‘pssh’ Boxes as outlined in [DMedia], Section 3.1.

A ‘pssh’ Box corresponds with a particular DRM if the `SystemID` field corresponds with that DRM’s ID as defined in [DSystem].

To add a license, the DECE Device SHALL:

1. Check for a DRM specific ‘pssh’ Box for the intended DRM
2. Create ‘pssh’ Box if missing
3. Add license to DRM specific ‘pssh’ Box, managing any pre-existing information in accordance with DRM rules (add to license acquisition information, add to pre-existing license, replace pre-existing license or acquisition information, etc.), and not exceeding the maximum size specified for each ‘pssh’ Box.

4. Adjust size of ‘free’ Box in ‘moov’ to prevent change of DCC size or reformat the DCC.

To remove a license, the DECE Device SHALL

1. Check for a DRM specific ‘pssh’ Box for the intended DRM, remove if necessary
2. If ‘pssh’ Box removed or changed, adjust size of ‘free’ Box in ‘moov’ to prevent change of DCC size.

8 Playing Content

This section describes the playback process.

Before a DECE Device can play a DCC, the following conditions must be met:

1. The DECE Device must be in a Domain
2. A valid DCC must be available to the DECE Device;
3. A valid license to the DCC bound to DECE Device's DRM Domain must be available to the DECE Device

DECE Devices MAY be pre-loaded with DCCs and Licenses at the time of Device purchase or manufacture.

Before a DECE Device can stream using Common Streaming, the following conditions must be met:

1. The DECE Device must be in a Domain
2. A valid MPD must be available to the DECE Device;
3. A valid license to Content referenced by the MPD bound to DECE Device's DRM Domain must be available to the DECE Device

A DECE Device SHALL comply with all the requirements of a Common Player as defined in Annex A, as modified by this section.

A DECE Device that supports Common Streaming SHALL comply with all requirements of a Common Streaming Player as defined in Annex B, as modified by this section.

"HD Device" and "SD Device" are defined in [DSystem], Section 4.7.4.

8.1 Media Profile, Codec and Delivery Target Support

A DECE Device is classified in terms of Interoperability Points which are a combination of Media Profile (e.g., SD, HD), Delivery Target (i.e., download or streaming; single-track, multi-track), and required codecs within the Media Profile (e.g., AVC and HEVC). Media Profiles ("Profile") and Delivery Targets ("Target") are described in [DMedia], Section 1.7.7 and defined in [DMedia] Annexes B and C respectively. Interoperability Points are defined in DMedia, Annex E.

Each Media Profile is associated with a set of picture formats, audio and video codecs, metadata, and other parameter values in the [DMedia]. To support any particular Media Profile, a DECE Device SHALL be able to handle all of the allowed format, codec and parameter options for that Profile.

Profile support is downwardly inclusive:

- An HD Device SHALL play Content as referenced in [DMedia], Annex E, Table E-2, “AVC Download and Prepackaged Interoperability Points”. Note that these are the download and prepackaged combinations.
- An HD Device MAY play Content as referenced in [DMedia], Annex E, Table E-4, “HEVC Download and Prepackaged Interoperability Points”.
- An HD Device MAY play Content as referenced in in [DMedia], Annex E, Table E-3, “AVC Streaming Interoperability Points”.
- An HD Device MAY play Content as referenced in in [DMedia], Annex E, Table E-5, “HEVC Streaming Interoperability Points”.
- If an HD Device is capable of playing any Streaming Interoperability Point, it SHALL play all Streaming Interoperability Points with Media Profile of “SD” and “HD”. For the avoidance of doubt, it must support both or neither.
- An SD Device SHALL play Content as referenced in [DMedia], Annex E Annex E, Table E-2, “AVC Download and Prepackaged Interoperability Points” with Media Profile of “SD”.
- An SD Device MAY play Content as referenced in [DMedia], Annex E Annex E, Table E-4, “HEVC Download and Prepackaged Interoperability Points” with Media Profile of “SD”.
- An SD Device MAY play Content as referenced in [DMedia], Annex E, Table E-3, “AVC Streaming Interoperability Points”, with Media Profile of “SD”.
- An SD Device MAY play Content as referenced in [DMedia], Annex E, Table E-5, “HEVC Streaming Interoperability Points”, with Media Profile of “SD”.

Profile support is upwardly exclusive:

- An SD Device SHALL NOT play Content with an HD Profile.

8.2 DCC Support

DECE Devices SHALL be able to decode and present all DCCs in accordance with Conditions for Playback as defined in A.1 plus the following condition:

- A valid DRM license consistent with the Device's Domain is available to the Device, possibly in the DCC as outlined in [DMedia], Section 3.1;

DECE Devices SHALL locate Licenses as defined in Section 7.2.5, License Management in DCC.

8.2.1 File Media Type and Filename Extension

Devices SHALL recognize files with Media Type (MIME type) or extension specified in [DMedia], Annex E, Section E.5.

Devices SHALL recognize files Media Type (MIME type) or extension as specified in [DDMP], Section 4.4.

8.2.2 DECE Version Compatibility

A DECE Device SHALL play DCCs that meet all of the following conditions:

- The DCC meets Common Player CFF Compatibility requirements defined in Annex A
- The DCC is compliant with DECE-specific constraints on the File Type Box ('ftyp') and Content Information Box ('cinf') as specified in [DMedia], Annex E, Section E.1.2 for a Compatible Version listed in [DMedia], Annex A.

8.3 Content Encryption

DECE Devices SHALL comply with Common Player Content Encryption requirements as defined in Section A.2.

8.4 Audio, Video and Subtitle Elementary Stream Requirements

DECE Devices SHALL comply with Common Player Audio, Video and Subtitle Elementary Stream Requirements as defined in Section A.3.

8.4.1.1 Subtitle Playback

DECE Devices SHOULD support rendering of all glyphs that correspond to the Unicode Code Points defined the "da", "nl", "fi", "fr", "de", "it", "no", "sv", “es” and “pt” Language Subtags in [DMedia], Section 6.9.2, Table 6-1. Note that this requirement will become normatively 'SHALL' in a future version of this specification.

8.5 Licensed Applications

A DRM Client in a DECE Domain SHALL NOT allow an unlicensed Licensed Application to decrypt DECE licensed Content.

8.6 Restrictions on Distributing DECE Device Functions

Although some functions may be distributed to Tethered Hosts and Device Proxies, playback may not. Playing Content function includes DRM Client's decryption function, decoding and output control functions including re-encrypting content using Approved Output Technologies.

DECE Devices SHALL NOT distribute Playing Content functionality to Tethered Hosts, Device Proxies, or Access Portals.

8.7 DECE Media Package (DMP) Playback Support

To playback Content from a DMP, DECE Devices must be able to locate and read tracks from multiple DCCs, typically in DMPs. Within a DMP, as per [DDMP] Media Presentations are defined in a TableOfContents part and Media Presentations are defined in Presentation parts of the DMP. Late Binding requires the ability to playback any combination of video track, audio track and subtitle track from a Media Presentation.

DECE Devices SHALL meet all requirements from Section 8.1 through 8.6 for playback of tracks from one or more DCCs in a DMP or equivalent as defined in [DDMP] and [DMedia]. A DMP equivalent refers to DMP information stored in a manner other than in a DMP, such as on Devices that do not support Outbound File Transfer.

DECE Devices MAY play one or more track via progressive download while playing one or more tracks from a DMP.

8.8 Common Streaming Playback

DECE Devices that support Common Streaming SHALL comply with Common Streaming Player requirements as defined in Annex B.

DECE Devices that support Common Streaming SHALL support Streaming Interoperability Points as defined in Section 8.1.

DECE Devices that support Common Streaming SHALL meet all requirements from all of the following, except as noted in this specification:

- Section 8.1 and 8.3 through 8.6 for playback
- [DSystem], Section 13.2.3

Note that the above references are in addition to those stated in Annex B.

Section 7 defines obtaining a Fulfillment Manifest including FufillmentManifest/StreamingInfo. Further defined was the method of obtaining BaseLocations that contains information necessary to acquire Licenses.

DECE Devices SHALL locate Licenses as defined in [DStream], Section 4.5.

8.9 Experience Media Application Playback

DECE Devices SHALL comply with Common Player Experience Media Playback requirements as defined in Section A.6.

9 User-Related Requirements

9.1 User Authentication

Devices SHALL manage Security Tokens in accordance with [DSecMech], Section 3.5.

9.2 Rights Locker Query and Display

9.2.1 Rights Query

DECE Devices MAY support Rights Query operations as defined in [DCoord] Section 7, and [DMeta], Section 3.

9.2.2 Rights Display

A DECE Device MAY display Rights information obtained from the DECE Device Portal.

Proportional scaling is a transformation on an image that changes the number of horizontal and vertical pixels in the same ratio within single-pixel rounding errors. Aspect ratio is maintained. Non-proportional scaling changes horizontal or vertical sizes in different ratios.

DECE Devices MAY proportionally scale images in a Container if such image is the desired aspect ratio but not the desired size. DECE Devices SHOULD scale larger images to a smaller images rather than scaling smaller to larger.

DECE Devices SHALL NOT crop images from a Container if an image of the desired aspect ratio is present in the Container. If a DECE Device crops an image it MAY also proportionally scale that image.

DECE Devices SHALL NOT scale non-proportionally.

9.3 Ratings Enforcement

Devices SHALL restrict Content playback based on ratings in DCCs. Ratings in DCCs is in Mandatory Metadata as defined in [DMedia] Section 2.3.4.

A DECE Device SHOULD restrict the display of Rights based on Rating information in Metadata associated with the Right (such as, metadata obtained from the Portal as part of the Rights query).

A Device MAY have a user-modifiable device-specific parental control setting.

Parental Control information can be obtained from the Coordinator using the Policy query mechanism defined in [DCoord], Section 5.6 using Parental Control Policies as defined in [DCoord], Section 5.5.3.

10 DLNA (Informative)

This section is for information purposes only.

It is envisioned that some DECE Devices will also be DLNA devices. In order for such a device to render content in a similar way as that defined in DLNA, DECE-related metadata needs to be placed in the DLNA Content Directory Service (CDS) in a standardized way. This section explains how a DLNA Digital Media Server (DMS) that serves UPnP AV CDS places such metadata into a CDS item that refers to a DCC.

Upon acquisition of a DCC, a DECE Device which also hosts a DLNA DMS or a UPnP MediaServer Device which supports ContentDirectory Service:3 [UPNPCDS3] or higher SHOULD create a CDS item which encapsulates the Required Metadata found in the DCC as defined in [DMeta], Section 4.1 in a *upnp:foreignMetadata* property; if it does so, it SHALL use the values indicated in the table below:

UPnP CDS Property	Value
upnp:foreignMetadata@type	"uvvu.com_mddece"
upnp:foreignMetadata::fmId	Value of mddece:APID
upnp:foreignMetadata::fmClass	"UltraViolet Container" + value of mddece:DECEMediaProfile
upnp:foreignMetadata::fmProvider	Value of mddece:Publisher
upnp:foreignMetadata::fmBody@xmlFlag	1
upnp:foreignMetadata::fmBody::fmEmbeddedXML	mddece:MetadataMovie including all child elements
dc:title	value of mddece:TitleDisplay60
res@duration	Value of RunLength converted to "H+:MM:SS" format
dc:date	Value of mddece:ReleaseDate converted to [ISO 8601] format
dc:description	Value of mddece:Summary190
res@protocolInfo	"http-get:*:video/vnd.dece.mp4:*"

The values of APID and DECEMediaProfile can be found in the 'ainf' box; all other metadata referenced in this table can be found in the 'meta' box in the DCC.

11 DECE Media Package (DMP) Support

The DECE Media Package (DMP) format allows one or more Digital CFF Containers (DCCs) to be stored, together with additional metadata and application data, in a single archive. The DECE Media Package is defined in [DDMP].

11.1 DMP Support

Devices complying with this specification support functionality associated with DMPs. Not all Devices are required to support all DMP functions, as indicated in this specification.

11.1.1General requirements

In the case that this specification has missed a particular case covering DMPs, this general requirement is provided.

DECE Devices SHALL support all DCC requirements, regardless of whether the DCC is part of a DMP or not; except where noted.

DECE Devices SHALL NOT peform any operation on a DMP that results in a DMP that does not comply with [DDMP]. That is, not operation can corrupt the DMP.

When using a DMP, DECE Devices SHALL NOT access the following from the DCC

- Base Locations (‘bloc’ Box)
- Licenses
- Metadata (‘meta’ Box).

11.1.2Primary Video Track

For licensing purposes, this specification defines the term Primary Video Track.

Each video track in each DCC within a Presentation has a type defined within that Presentation’s Mandatory Metadata by the MetadataMovie/Track/Video/Type element. Note that the association is through MetadataMovie/Track/Video/TrackReference. If a video track’s Type=’primary’ or the Type element is absent (that is, default is ‘primary’ as per [TR-META-CM]), then it is the Primary Video Track.

A Presentation SHALL have only one Primary Video Track.

Note that currently, a Presentation has only one video track. However, it is not allowed to assume this is the Primary Video Track as other video tracks might be added in future implementations. For backwards compatibility it is important that implementation properly recognize the Primary Video Track. An implementation that does not distinguish the Primary Video Track from other video tracks is non-compliant.

12 File Management

12.1 Default File Locations

This section specifies where Licensed Applications are encouraged to store DECE files. In this context, a DECE file is a DCC or DMP stored in a file system.

There are two categories of devices where it is applicable for Licensed Applications to store files in a specific location

- Devices that are capable of Outbound File Transfer, including devices where Licensed Applications are capable of writing removable media
- Devices or collection of devices where multiple Licensed Applications can share a file system. Note that the file system might be internal such as on a general purpose computer, or external, such as on a Network Attached Storage (NAS) device.

Although there might be other conditions where writing files to a known location applies, this section assumes that files will be written in specific file locations given those conditions.

12.1.1Using Default File Locations

As applicable, Licensed Applications SHALL comply with requirements defined in Annex C, where Annex C “Applications” are Licensed Applications, and Annex C “ecosystem files” in are DECE files.

Although these requirements are written towards Licensed Applications, they are strongly recommended for all other entities accessing DECE files such as Download Managers that are independent of Licensed Applications.

12.1.2Default File Locations

Licensed Applications SHALL use Default File Location as specified inAnnex C. where <ecosystem> is ‘UltraViolet’ and <ecosystem-short> is ‘UV’.

Annex A Common Player

This section defines a set of requirements collectively referred to as a "Common Player".

The intent is for other sections of this document or documents to reference this Annex and supplement these requirements as appropriate.

A.1. Conditions for Playback

Common Players SHALL be able to decode and present all DCCs under the following conditions:

- The DCC's Media Profile (e.g., SD or HD) is supported by the Common Player
- The DCC's Delivery Target is supported by the Common Player
- Content protection rules are met
- CFF Compatibility rules are met, as defined in Section A.1.1
- The DCC is valid as per all relevant ecosystem specifications

Note that since DCC are ISO File Format compliant, additional boxes not specified in [DMedia] can be present in the DCC.

Client Implementers should note that encoding rates and allowable numbers of tracks can result in DCC sizes larger than 2^32 bytes (~4 GB), so therefore Device file systems will need to support files of that size.

A.1.1. CFF Compatibility

The DCC is considered CFF Compatible if contains a File Type Box ('ftyp') in accordance with [DMedia], Section 2.3.1 and a Content Information Box ('cinf') in accordance with [DMedia], Section 2.2.2.

A.2. Content Decryption

Common Players SHALL be able to decrypt content using AES CTR Mode as defined in [DMedia], Section 3.

A.3. Audio, Video and Subtitle Elementary Stream Requirements

Full details of the audio and video codecs and how the corresponding elementary streams are placed in the DCC can be found in [DMedia].

Common Players that play Content of a given Media Profile and Delivery Target SHALL play Content specified in [DMedia] Annex C, corresponding with that Delivery Target, with constraints imposed on that Media Profile.

Common Players NEED NOT support 25 Hz or 50 Hz Content as defined in [DMedia] Tables C-4 and C-6.

A.3.1. Audio Requirements

Common Players SHALL decode and present audio as defined in the [DMedia], Section 5.

Common Players SHALL decode and present the Selected Audio Track in accordance with Section A.3.4.

A.3.1.1. AAC LC Stereo Support

Common Players SHALL be able to decode AAC LC stereo audio as defined in the [DMedia], Section 5.3.2.

Common Players SHALL be capable of decoding MPEG-4 AAC LC stereo content at bit rates 320 kbps or less, and that were encoded at a sample rate of 44.1 kHz.

Note that this requirement is intended to assist backward compatibility of devices with future DECE versions that include music-only media files.

When decoding AAC LC stereo audio, Common Players NEED NOT interpret the following MPEG-4 AAC elementary stream elements:

- fill_element (FIL)
- data_stream_element (DSE)

A.3.1.2. Other Audio Codecs

The DCC also supports other optional audio codecs.

Common Players MAY implement any Audio CODEC from the [DMedia], Section 5 as long as it is consistent with the Media Profiles and Delivery Targets supported by that Common Player.

A.3.1.3. Audio Downmixing

If decoding a multi-channel audio track to an output supporting fewer channels, the Common Player SHALL downmix to the available output channels according to the audio codec recommendations.

For example, when playing a 5.1 channel mix on a 2-channel output, 5.1 channels is downmixed to 2 channels.

A.3.1.4. Output of Encoded Audio

If a Common Player is driving a digital audio output (e.g. SPDIF, HDMI, etc.) that supports the transport of an encoded audio, then the Common Player SHALL be able to pass-through a multi-channel codec other than AAC to the audio output. This includes minor transport conversions necessary to convert from the DCC packaging to the output port packaging.

A.3.2. Video Requirements

Common Players SHALL decode and present video as defined in the [DMedia], Section 4.

Common Players SHALL support scaling in a manner that supports subsampling as defined in [DMedia].

A.3.3. Subtitles

The following sections include requirement for subtitles. Note that this includes what are commonly referred to as ‘captions’.

For this section, for a subtitle track, Subtitle Track Language is defined as the value of MetadataMovie/TrackMetadata/Track/Subtitle/Language (as specified in [DMeta], Section 4) corresponding with the subtitle track.

A match is found between Subtitle Track Language and ‘Language Subtags’ of [DMedia], Section 6.9, Table 6-1 by performing a Lookup ([RFC4647], Section 3.4) of the Subtitle Track Language with all Language Subtags in [DMedia], Section 6.9.2.2, Table 6-1 as the language priority list ([RFC4647], Section 2.3). If this operation results in ‘default’, then no match is found.

Common Players SHALL decode and present both text profile and image profile subtitles as per [DMedia] v1.0.7, and all versions adopted up to the DMedia specification defined by [DMedia] Annex A {DMEDIA_VERSION_POINTS}. If the schema location field of SubtitleSampleEntry in the CFF-TT subtitle track Sample Description Box (‘stsd’) does not conform to [DMedia] v1.0.7 or any version adopted up to the DMedia specification defined by [DMedia] Annex A {DMEDIA_VERSION_POINTS}, Common Players SHALL ignore the value of the schema location field of SubtitleSampleEntry.

Note: see also [DSystem] Section 1.6 for requirements related to XML Change Management.

A.3.3.1. Subtitling Track Selection

Common Players SHALL determine Primary Subtitling Presentation Track, as defined in [DMeta] in accordance with Section A.3.4.

Common Players SHALL determine Alternate Subtitling Presentation Track, as defined in [DMeta] in accordance with Section A.3.4.

For purpose of track selection, a text track SHALL NOT be considered playable if both of the following are true:

- the Subtitle Track Language matches a 'Language Subtags' of [DMedia], Section 6.9.2, Table 6-1
- the Common Player does not support rendering of all glyphs that correspond to the Unicode Code Points defined for matching Language Subtag in [DMedia], Section 6.9.2, Table 6-1

A Common Player SHALL NOT select tracks that are not playable.

A.3.3.2. Subtitling Mode

A Common Player SHALL be in either Primary Subtitling Presentation Mode or Alternate Subtitling Presentation Mode.

As an initial condition, unless otherwise specified, a Common Player SHALL be in Alternate Subtitling Presentation Mode.

If a Primary Subtitling Presentation Track is selected through the Subtitling Track Selection process (Section A.3.3.1), the Common Player SHALL be in the Primary Subtitling Presentation Mode, unless the User has opted to turn subtitles off. That is, when a Primary Subtitling Presentation Track is selected, the Device is in Primary Subtitle Presentation Mode unless the User selects otherwise.

When a Primary Subtitling Presentation Track is selected through the Subtitling Track Selection process (Section A.3.3.1), Common Players SHOULD provide the means for a User to turn subtitles off (Alternate Subtitling Presentation Mode) and on (Primary Subtitling Presentation Mode).

A.3.3.3. Subtitle Playback

During playback, when in Primary Subtitling Playback Mode, the Common Player SHALL decode and present the Primary Subtitling Mode Track.

During playback, when in Alternate Subtitling Playback Mode and an Alternate Subtitle Presentation Track has been selected through the Subtitling Track Selection process (Section A.3.3.1), Common Players SHALL decode and present forced elements as per [DMedia] Section 6 from the Alternate Subtitle Presentation Track. In the case of a Forced Subtitle Track, this is all elements.

Common Players SHALL NOT decode and present more than one subtitle track simultaneously.

Common Players SHALL decode and present subtitles as per [DMedia], Section 6.

Note that Devices process both text profile and image profile subtitles.

Common Players SHALL render Graphic and Format Unicode Code Points encountered in textual content within CFF-TT subtitles. The Glyph rendered for Graphics and Format Unicode Code Points SHALL correspond with the character represented by the Unicode Code Point as per [UNICODE] or, in the case where the Unicode Code Point is not supported by the Common Player, the Glyph rendered SHALL correspond with Unicode Code Point U+25A1 ("WHITE SQUARE") or Unicode Code Point U+005F ("LOW LINE").

Common Players SHALL ignore Control Codes Unicode Code Points (U+0000..U+001F and U+007F..U+009F) encountered in textual content within CFF-TT subtitles. An ignored Control Code does not affect presentation. No glyph is rendered for a Control Code.

Common Players SHALL support rendering of all glyphs that correspond to the Unicode Code Points defined for the “x-ALL” Language Subtag sequence in [DMedia], Section 6.9.2, Table 6-1. Note that this includes support for ‘en’ (English).

Common Players SHALL satisfy the observed presentation behavior of subtitle rendering model defined in [DMedia], Section 6, including the behavior seen with the minimum performance requirements defined in [DMedia] Annex B. Common Players are allowed to use any implementation that satisfies the observed behavior defined by the model and are allowed to have higher decoding, drawing or text rendering rates.

Common Players are allowed to render/paint and scale to different resolutions than the SMPTE TT root container in order to optimize for presentation to the display connected to (or integrated as part of) the Common Player, provided that they maintain relative position between subtitles and video. Common Players SHALL maintain subtitle and video relative position (regardless of differences in resolution between the display and SMPTE TT root container).

Common Players SHALL composite subtitles on to a minimum of 16-bit full color Subtitle Plane as per [DMedia] Section 6.

Common Players SHALL match the Subtitle Plane and Video Plane color space for subtitle overlay.

For random access or subtitle track switching, Common Players SHOULD search for the subtitle fragment that includes the composition time for the random access video sample, and prepare subtitles for presentation from the random access point into the video presentation.

Note that Media Clients will need to acquire the ‘mfra’ Box at the end of the file to properly random access subtitle fragments.

A.3.4. Audio and Subtitle Track Selection

The purpose of this section is to define behavior for Devices with respect to audio and subtitle track selection. This allows Devices to select default tracks consistent with the intent of the Content Providers.

Information is provided to assist Devices in selecting the appropriate audio and subtitle information. Information is contained in MetadataMovie/TrackMetadata and MetadataMovie/TrackSelections, both defined in [DMeta].

Track selection is made based on Device defaults (e.g., region and language), User preferences (e.g., language, accessibility), available tracks, TrackMetadata information and, if available, TrackSelections information. Generally, Users may override Device track selection. However, there are no overrides for Alternate Subtitle Presentation Track.

Additional terminology used in this section is defined in [DMeta], Section 5.

A.3.4.1. Default Track Selection

Container Metadata provides information that can be used to select default tracks in accordance with the Content Provider’s intent. These data are defined in [DMeta], Section 4.1.5, and the assumed algorithm in [DMeta] Section 5.

System Language and other language preferences SHALL be at least one language that can be represented as Language Tags as per [RFC5646] from the IANA Language Subtag Registry [IANA-LANG].

Common Players SHALL set the System Language.

Common Players SHOULD set the default System Language to match the user Interface language or Operating System language.

Common Players SHOULD provide the means for Users to set the System Language.

Common Players SHOULD provide System Language settings to include dialects. For example, “Latin Spanish” (IANA tag ‘es-419’) should be distinguished from “Castilian Spanish” (IANA tag ‘es’).

Common Players SHOULD provide the means for User to select an audio preference for Type compatible with MetadataMovie/TrackMetadata/Track/Audio/Type. For example, the User can select a Type of ‘dialogcentric’.

Common Players SHALL provide the means for User to select a subtitle preference for Type compatible with MetadataMovie/TrackMetadata/Track/Subtitle/Type. For example, the User can select a Type of ‘SDH’.

Common Players SHALL provide the means for User to select a language preference for audio.

Common Players SHALL provide the means for User to select a language preference for subtitles.

Common Players SHOULD provide the means for User to select a preference dubbed audio or original audio.

Prior to playback, Common Players SHALL use track selection methods described in [DMeta], Section 5 to select default audio and subtitle tracks.

A.3.4.2. User Track Selection

Common Players SHALL provide the ability for a User to select the Selected Audio Track.

Common Players SHALL provide the ability for a User to select the Primary Subtitling Presentation Track.

Common Players SHALL NOT allow a User to select a Forced Subtitle as the Primary Subtitling Presentation Track.

Common Players SHALL NOT allow the User to select the Alternate Subtitling Presentation Track.

Common Players MAY persistently store User track selections for later use. In the future, these can be used in lieu of default track selection for this DCC.

A.3.5. Synchronization

Common Players SHALL synchronize audio and video presentation within 15-millisecond accuracy.

Common Players SHALL synchronize subtitle and video presentation with frame accuracy.

Common Players SHALL support an Edit List Box with a Timeline Mapping Edit entry as described in [DMedia] 2.4, and if included, shift the media timeline relative to the presentation timeline so that the specified Media-Time in the media timeline is mapped to the start of the presentation timeline.

Note: If an Edit List Box exists, inter-track synchronization is based on the Device appropriately processing the Edit List Box and mapping the media timeline to the presentation timeline.

A.3.6. Chapters

When random accessing a chapter from EntryTimecode (as defined in [DMeta] 4.1.4), Common Players SHOULD start presentation at the video frame at or immediately following that time on the movie presentation timeline. For the avoidance of doubt, if the referenced picture is not a video random accessible sample, decoding may need to begin at a previous random accessible sample and prior pictures not presented.

When Common Players begin playback from a Chapter time, they SHOULD start Audio presentation on the next sync sample on the movie timeline at or after the first video frame presented.

When Common Players begin playback from a Chapter time, they SHOULD start Subtitle presentation with the next Subtitle Event on the movie timeline at or after the first video frame presented.

For avoidance of doubt, as with all playback, video, subtitle, and audio synchronization to the movie presentation timeline is to be maintained.

Common Players MAY display chapter images as referenced in the Chapter element.

Common Players MAY display text as included in the Chapter/DisplayLabel element.

If multiple instances of DisplayLabel are provided, the Common Player SHOULD determine the best language as if the DisplayLabel element is a Subtitle track (i.e., DisplayLabel/@language is equivalent to Track/Subtitle/Language element). For example, if DisplayLabel and Subtitle have the same languages, DisplayLabel will have the same language as subtitles.

Common Players MAY create and display their own text or images for chapter selection; for example, “Chapter 1”.

A.4. Trick Play

Common Players MAY be capable of trick play. Examples of trick play are fast forward, rewind and skip.

A.5. Common Media Package (CMP) Playback Support

To playback Content from a CMP, Common Players must be able to locate and read tracks from multiple DCCs, typically in CMPs. Within a CMP, as per [DDMP] Media Presentations are defined in a TableOfContents part and Media Presentations are defined in Presentation parts of the CMP. Late Binding requires the ability to playback any combination of video track, audio track and subtitle track from a Media Presentation.

Common Players SHALL meet all requirements from Section A.1 through A.4 for playback of tracks from one or more DCCs in a CMP or equivalent as defined in [DDMP] and [DMedia]. A CMP equivalent refers to CMP information stored in a manner other than in a CMP, such as on Devices that do not support Outbound File Transfer.

Common Players MAY play one or more track via progressive download while playing one or more tracks from a CMP.

A.6. Experience Media Application Playback

The Experience Media Application is a DMP Media Application that defines the relationship between various media objects, allowing a Device to allow the User to navigate Content.

Experience Media Application is structurally defined [DDMP], Section 4.3.7.1 and functionally defined in [DCManifest].

Any Experience Media Application functions implemented by a Common Player SHALL be implemented in accordance with [DDMP], Section 4.3.7.1 and [DCManifest]. In case of conflict, precedence is this specification, then [DDMP], then [DCManifest]. That is, this specification supersedes the other specifications.

A.6.1. General Requirements

A Common Player SHALL have the ability to parse the Experience Media Application for the purpose of playing all Content referenced by Experience elements.

A Common Player SHALL provide the ability for a User to play any Audiovisual Content referenced within Experiences.

A.6.2. Navigation

Common Players SHALL provide the User the ability to navigate Experiences. Note that the starting Experience element is the first instance of Experience in ExperienceMediaApp/Experiences.

Common Players SHALL present metadata associated with each Experience element or Experience/Audiovisual element as it is navigated.

A.6.3. Sequential Playback

The Media Manifest has provisions to play a single Presentation, play a sequence of clips (Presentation subsets) and/or images. Devices are required to support all of these playback models.

A Common Player SHALL playback Presentations referenced in Experience/Audiovisual/PresentationID. Note that his is within an Audiovisual element. Note that the order of Audiovisual elements within an Experience has no meaning.

A Common Player SHALL playback Clips and Image Clips referenced in Experience/Audiovisual/Presentation/PlayableSequence as defined in [DCManifest], Section 5.2.3 with the following conditions

- Common Players SHALL play Clips and Image Clips in accordance with their respective @sequence elements.
- Presentation is referenced in PlayableSequence/Clip/PresentationID
- Timecodes are relative to the Presentation time, with 0 being the beginning of a Presentation or clip.
- Common Players SHALL support Clip playback in accordance with Clip/EntryPointTimecode starting within 1 frame of the specified time. Note that the absence of Clip/EntryPointTimecode requires the Device to start playback at the beginning of the Clip
- Common Players SHALL support Clip playback in accordance with Clip/ExitPointTimecode ending within 1 frame of the specified time. Note that the absence of Clip/ExitPointTimecode requires the Device to stop playback at the end of the Clip.
- Common Players SHALL play Clips and Image Clips are in the order defined by @sequence. If @sequence is absent and more than one Clip exists, the Device SHOULD play Clips in the order in the PlayableSequence element.
- If the @seamless attribute is missing or 'false', Common Players SHALL present no more than 1 second of black and silence between the playback of the referenced content in each Clip or Image Clip.
- If the @seamless attribute is present and 'true', Common Players SHALL playback Clips and Image Clips without delay. That is, Clips must play seamlessly. Process for seamless playback is at the discretion of the player, although constraints could be provided in a future specification.

A.6.4. Galleries

Galleries are collections of images.

Common Players SHOULD present Galleries.

Annex B Common Streaming Player

This section defines the requirements for a Common Streaming Player.

B.1. Common Streaming Playback

Note that Section 7 defines obtaining a Fulfillment Manifest including FufillmentManifest/StreamingInfo. Further defined was the method of obtaining BaseLocations that contains information necessary to acquire Licenses.

B.1.1. General Requirements for Streaming

Common Streaming Players as defined in [DSystem] have the capability to playback tracks streamed from a network.

Common Streaming Players SHALL support a Streaming Interoperability Points as defined in Section 8.1.

Common Streaming Players SHALL meet all requirements from all of the following, except as noted in this specification:

- Section A.1 through A.3 for playback
- [DStream]

B.1.2. Common Streaming Data Sources

Common Streaming Players, when streaming SHALL obtain information for Default Track Selection from StreamingInfo/ExperienceStreamApp/Presentations.

Common Streaming Players, when streaming SHALL obtain the MPD used for Common Streaming Protocol as defined in [DStream] from StreamingInfo//MPDLocation.

B.1.3. Common Streaming Experience

Common Streaming Players, when streaming SHALL support Experience Media Application Playback as defined in Section 8.9 using information in StreamingInfo/ExperienceStreamApp.

B.1.4. Common Streaming Protocol Requirements

Common Streaming Players SHALL support streaming protocol in accordance with [DStream].

B.1.5. Streaming Media Playback Requirements

This section is equivalent to Section 8.2, but for Common Streaming.

Devices SHALL be able to decode and present all Common Streaming streams under the following conditions:

- A valid DRM license consistent with the Device's Domain is available to the Device, as defined in [DStream], Section 4.5;
- The stream's Media Profile (e.g., SD or HD) is supported by the Device;
- The stream's Delivery Target is supported by the Device;
- The stream conforms to DMedia v1.2 or any versions adopted up to the DMedia specification defined by [DMedia] Annex A {DMEDIA_VERSION_POINTS}.
- Content protection rules are met
- The stream is valid as per all relevant DECE specifications.

B.1.5.1. Content Encryption

Devices SHALL be able to decrypt content using AES CTR Mode as defined in [DMedia], Section 3.

B.1.6. Specific Playback Environments

B.1.6.1. HTML5

One model for DASH is to implement the Player, including download protocol in HTML/ECMAScript. This model is valid for DECE, although the HTML/ECMAScript is not defined by DECE. For the implementation to be considered Common Streaming, it must comply with the requirements for Common Streaming (e.g., [DStream]).

Devices MAY support Common Streaming through an HTML5 browser.

B.1.6.2. Combined Streaming and Playback from DCCs

Common Streaming Players SHALL play tracks entirely via streaming. That is, Players need to be capable of playing Content entirely through the Common Streaming mechanism (i.e., no download).

Common Streaming Players MAY play one or more track via streaming while playing one or more track from a DMP.

B.1.6.3. Adaptive Progressive Download

Adaptive Progressive Download is a hybrid of Progressive Download and Common Streaming. Although it incorporates Common Streaming concepts, it is still a download operation as the end result is downloaded DCC.

When performing Adaptive Progressive Download a Device performs the following functions

- Download an adaptation set whose highest bitrate video track is a CFF single track DCC and lower bitrate tracks are CSF or CFF
- Progressive download can be any segment from the adaptation set
- Following progressive download, any segment that is not highest bitrate is replaced.

Common Players MAY play using 'Adaptive Progressive Download', when available.

Annex C Default File Storage and Locations

This section describes how and where media files are stored. This section uses the term 'Application' to refer to any application involved in writing media files. Ecosystem files are DCCs, DMPs and any other files involved in playback in the applicable ecosystem.

All requirements in this section assume the feasibility of writing files to the locations indicated. That is, the requirements are irrelevant if it is not feasible for the Application to write to or read from the specified location as applicable.

The term Default File Location is defined in 0

C.1. Using Default File Locations

C.1.1. Storing Files

Applications storing ecosystem files on a local file system or removable media SHOULD by default store ecosystem files at the applicable Default File Location.

Applications storing ecosystem files on a local file system or removable media SHALL provide Users the ability to select the Default File Location as the default location for storing ecosystem files.

If the Application intends to store an ecosystem file at the Default File Location, but that location does not exist, the Application SHALL create the location associated with the Default File Location, if possible.

C.1.2. Locating Files

Applications searching for ecosystem files on a local file system or removable media SHOULD by default first search for ecosystem files at the applicable Default File Location.

Applications accessing ecosystem files from a local file system or removable media SHALL provide Users the ability to select the Default File Location as the default location for searching for ecosystem files.

C.1.3. Setting Location for Reading and Storing Files

Applications that do not by default store files in the Default File Location SHALL provide a means for the User to change the storage location to the Default File Location, where feasible.

Applications that do not by default search for files in the Default File Location SHALL provide a means for the User to add the Default File Location to the search path, where feasible.

In multi-user environments, Applications MAY provide an option to store files in the "public" folder to make them discoverable by other users.

C.2. Default File Location

Default File Locations identify for a given operating system and/or the storage medium the default location for media files. This section indicates the correct Default File Location for various operating systems and for removable storage.

The symbols <ecosystem> and <ecosystem-short> must be defined for each ecosystem. For example, for DECE, <ecosystem> would be 'UltraViolet' and <ecosystem-short> would be 'UV'.

C.2.1. Microsoft Windows

The Default File Location for Microsoft Windows operating systems is not a fixed location. Rather it is obtained using the following steps:

- Default path for Windows 7 and newer is: `%USERPROFILE%\Videos\`<ecosystem>, derived as follows:
  - Pass `FOLDERID_VideosLibrary` to `IKnownFolder` to determine default video save location.
- Default File Location for Windows versions older than Windows 7 is: `%USERPROFILE%\My Documents\My Videos\`<ecosystem>, constructed as follows:
  - Use `FOLDERID_Videos` if available, otherwise use `CSIDL_VIDEOS` if available
  - If GUIDs not available, use legacy default path below.
  - If `%USERPROFILE%` not available substitute null and use default drive

C.2.2. Apple Mac OS X

The Default File Location for Apple Mac OS X is: `/Users/<username>/Movies/`<ecosystem> where `<username>` is the Mac OS username.

C.2.3. Unix, Linux, Posix and other Unix variants

The Default File Location of Unix, Linux, Posix and other Unix variants is: `/home/media/`<ecosystem>

C.2.4. Android OS

- For API Level 8 or above, Default File Location is determined as follows: `getExternalStoragePublicDirectory(DIRECTORY_MOVIES)`, create <ecosystem> subdirectory if not present
- For APIs prior to API Level 8, the Default File Location is `/sdcard/Movies/`<ecosystem>. If not available, for Default File Location use: `/Movies/`<ecosystem>

C.2.5. General Media (FAT, NTFS, HFS, etc.; or NAS, etc.)

General Media refers to storage such as other fixed media, removable storage media (such as, USB, SD card, hard disk, optical discs, etc.), and external media such as NAS. File location is independent of operating system. A hierarchical file system (i.e., one with a top level directory and subdirectories) is assumed.

The Default File Location is: <root>`Movies`<path separator><ecosystem>

Where

- <root> is the indicator of the root directory; for example "/" or "\". Where practical, root SHOULD be a shared location, rather than a user-specific path. An example of impractical is where multiple users to not have write privileges to a shared location.
- <path separator> is the character or characters used to separate directory levels in a path, typically "/" or "\".

The term <ecosystem-short> SHALL be substituted for <ecosystem-short> in file systems where that substitution would convert an invalid Default File Location into a valid Default File Location.

C.2.6. iOS

- iOS does not currently have a default location.

END