



Digital Music, Photo, and Video Collections



MPV CE Playlists Specification

**Revision 0.41
Working Draft**

18 November 2003

© 2003 Optical Storage Technology Association

IMPORTANT NOTICE

This document is a working draft for review by OSTA members and approved parties. It is a draft document and will be updated, replaced, or obsoleted by other documents at any time and without notice. It is inappropriate to use OSTA Working Draft documents as reference materials, to cite them in other publications, or to refer to them as anything other than a “work in progress”.

NOT FOR DISBTRIBUTION ON A PUBLIC WEBSITE

This document is available at <http://www.osta.org/mpv/mpvmbrs/specs/MPV-CEPlaylists-0.41WD.PDF>

POINTS OF CONTACT

<p><u>OSTA</u> David Bunzel OSTA President</p> <p>Tel: +1 (408) 253-3695 Email: dbunzel@osta.org</p> <p>http://www.osta.org</p> <p><u>MPV Website</u> http://www.osta.org/mpv/</p>	<p><u>Technical Content</u></p> <p>Pieter van Zee MPV Initiative Lead Editor, MPV Specification</p> <p>Tel: +1 541-715-8658 Email: Pieter_van_Zee@hp.com</p> <p>Felix Nemirovsky Chairman, MultiRead Subcommittee</p> <p>Tel: +1 415 643 0944 Email: felixn@pacbell.net</p>
---	---

ABSTRACT

The MPV CE Playlists Profile Specification tightly defines the MPV implementation practices and metadata required to guarantee interoperability of music, photo, and video playlists on consumer electronics products.

COPYRIGHT NOTICE

Copyright 2003 Optical Storage Technology Association, Inc.

REVISION HISTORY

Revision	Date	Comments
0.10	5 Feb 2003	1 st draft
0.20	1 March 2003	2 nd draft
0.30	15 March 2003	3 rd draft
0.40	21 March 2003	4 th draft
0.41	18 Nov 2003	brief review and minor updates around logo usage

LICENSING IMPORTANT NOTICES

- (a) THIS DOCUMENT IS AN AUTHORIZED AND APPROVED PUBLICATION OF THE OPTICAL STORAGE TECHNOLOGY ASSOCIATION (OSTA). THE SPECIFICATIONS CONTAINED HEREIN ARE THE EXCLUSIVE PROPERTY OF OSTA BUT MAY BE REFERRED TO AND UTILIZED BY THE GENERAL PUBLIC FOR ANY LEGITIMATE PURPOSE, PARTICULARLY IN THE DESIGN AND DEVELOPMENT OF WRITABLE OPTICAL SYSTEMS AND SUBSYSTEMS. THIS DOCUMENT MAY BE COPIED IN WHOLE OR IN PART PROVIDED THAT NO REVISIONS, ALTERATIONS, OR CHANGES OF ANY KIND ARE MADE TO THE MATERIALS CONTAINED HEREIN.
- (b) COMPLIANCE WITH THIS DOCUMENT MAY REQUIRE USE OF ONE OR MORE FEATURES COVERED BY THE PATENT RIGHTS OF AN OSTA MEMBER, ASSOCIATE OR THIRD PARTY. NO POSITION IS TAKEN BY OSTA WITH RESPECT TO THE VALIDITY OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT, WHETHER OWNED BY A MEMBER OR ASSOCIATE OF OSTA OR OTHERWISE. OSTA HEREBY EXPRESSLY DISCLAIMS ANY LIABILITY FOR INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS OF OTHERS BY VIRTUE OF THIS OSTA DOCUMENT, NOR DOES OSTA UNDERTAKE A DUTY TO ADVISE USERS OR POTENTIAL USERS OF OSTA DOCUMENTS OF SUCH NOTICES OR ALLEGATIONS. OSTA HEREBY EXPRESSLY ADVISES ALL USERS OR POTENTIAL USERS OF THIS DOCUMENT TO INVESTIGATE AND ANALYZE ANY POTENTIAL INFRINGEMENT SITUATION, SEEK THE ADVICE OF INTELLECTUAL PROPERTY COUNSEL AND, IF INDICATED, OBTAIN A LICENSE UNDER ANY APPLICABLE INTELLECTUAL PROPERTY RIGHT OR TAKE THE NECESSARY STEPS TO AVOID INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHT. OSTA EXPRESSLY DISCLAIMS ANY INTENT TO PROMOTE INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHT BY VIRTUE OF THE EVOLUTION, ADOPTION, OR PUBLICATION OF THIS OSTA DOCUMENT.
- (c) ONE OR MORE PATENT HOLDERS HAVE FILED STATEMENTS OF WILLINGNESS TO GRANT A LICENSE, ON REASONABLE AND NONDISCRIMINATORY TERMS, ON A RECIPROCAL BASIS, UNDER PATENT CLAIMS ESSENTIAL TO IMPLEMENT THIS SPECIFICATION. FURTHER INFORMATION MAY BE OBTAINED FROM OSTA.
- (d) OSTA MAKES NO REPRESENTATION OR WARRANTY REGARDING ANY SPECIFICATION, AND ANY COMPANY USING A SPECIFICATION SHALL DO SO AT ITS SOLE RISK, INCLUDING SPECIFICALLY THE RISKS THAT A PRODUCT DEVELOPED WILL NOT BE COMPATIBLE WITH ANY OTHER PRODUCT OR THAT ANY PARTICULAR PERFORMANCE WILL NOT BE ACHIEVED. OSTA SHALL NOT BE LIABLE FOR ANY EXEMPLARY, INCIDENTAL, PROXIMATE OR CONSEQUENTIAL DAMAGES OR EXPENSES ARISING FROM THE USE OR IMPLEMENTATION OF THIS DOCUMENT. THIS DOCUMENT DEFINES ONLY ONE APPROACH TO COMPATIBILITY, AND OTHER APPROACHES MAY BE AVAILABLE IN THE INDUSTRY.
- (e) THIS DOCUMENT IS A SPECIFICATION ADOPTED BY OSTA. THIS DOCUMENT MAY BE REVISED BY OSTA AT ANY TIME AND WITHOUT NOTICE AND USERS ARE ADVISED TO OBTAIN THE LATEST VERSION. IT IS INTENDED SOLELY AS A GUIDE FOR ORGANIZATIONS INTERESTED IN DEVELOPING PRODUCTS WHICH CAN BE COMPATIBLE WITH OTHER PRODUCTS DEVELOPED USING THIS DOCUMENT. THIS DOCUMENT IS PROVIDED "AS IS".
- (f) THE MPV NAME, THE MPV LOGO, AND THE MusicPhotoVideo NAME ARE TRADEMARKS OF OPTICAL STORAGE TECHNOLOGY ASSOCIATION, INC. ALL OTHER TRADEMARKS ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. THESE OSTA TRADEMARKS MAY NOT BE USED EXCEPT FOR JOURNALISTIC PURPOSES WITHOUT AN EXPLICIT LICENSE FROM OSTA.

Contents

Contents.....	4
Table of MPV CE Playlists Requirements.....	6
Chapter 1: Introduction.....	9
1.1 Executive Summary.....	9
1.2 MPV CE Playlists – Guaranteed Navigation, Enhanced Playback Experience.....	10
1.3 No Content Playback Guarantee.....	11
1.4 MPV Logo Program.....	11
1.5 MPV Profiles.....	11
1.6 Terms of Use.....	12
Chapter 2: MPV CE Playlists Profile 1.0.....	13
2.1 MPV CE Playlists Introduction.....	13
2.2 MPV CE Playlists Specification Practices.....	13
2.3 Formalities For Use of the MPV CE Playlists Profile.....	14
Chapter 3: MPV CE Playlists – Filesystem Requirements.....	16
3.1.1 Filesystem-based Asset Access.....	16
Chapter 4: MPV CE Playlists – Manifest Spec Practices.....	17
4.1 Practices for MPV Document spec.....	17
4.1.1 mpvd:writenBy attribute.....	17
4.1.2 mpvd:documentIdRef attribute.....	17
4.1.3 Locale of the Manifest.....	18
4.1.4 MPV Document Profile.....	18
4.2 Example.....	19
Chapter 5: MPV CE Playlists – Core Spec Practices.....	21
5.1 Practices for Asset Dublin Core metadata.....	21
5.1.1 dc plain-text elements: creator, description, identifier, publisher, rights, title.....	21
5.1.2 dc:format element.....	22
5.1.3 dc:language element.....	22
5.1.4 dc:subject element.....	23
5.1.5 dcterms:created and :modified elements.....	23
5.1.6 dc metadata use.....	23
5.2 Well-Known Music Asset Types.....	24
5.2.1 Top-level mpv:Audio.....	24
5.2.2 Primary mpv:Audio.....	24
5.3 Well-Known Photo Asset Types.....	24
5.3.1 Top-level mpv:Still.....	25
5.3.2 Primary mpv:Still.....	25
5.3.3 Thumbnail Rendition mpv:Still.....	26
5.3.4 Screen Rendition mpv:Still.....	26
5.3.5 Artwork Related mpv:Still.....	26

5.3.6	Top-level mpv:StillWithAudio.....	27
5.3.7	Top-level mpv:StillMultishotSequence	27
5.3.8	Alt Rendition mpv:StillMultishotSequence	27
5.3.9	Top-level mpv:StillPanoramaSequence.....	28
5.4	Well-Known Video Asset Types	28
5.4.1	Top-level mpv:Video.....	29
5.4.2	Show Rendition mpv:Video	29
5.5	Practices for Asset Relationships	29
5.5.1	Asset Relationships	29
5.5.2	Master asset.....	30
5.5.3	Screen rendition	30
5.5.4	Thumbnail rendition	30
5.5.5	Show rendition: TBD.....	31
5.5.6	Example:.....	31
5.5.7	Alt rendition.....	31
5.5.8	Subsampled rendition	32
5.5.9	DerivedFrom Related Asset	32
5.5.10	CopyOf Related Asset	32
5.6	Practices for Asset mpv:LastURLs	32
5.6.1	mpv:filesystem attribute	32
5.6.2	UTF-8 and URL-encoding.....	33
5.6.3	Fragment Identifier Processing	33
5.7	Practices for mpv:Still.....	33
5.7.1	Image Size metadata.....	33
5.7.2	mpv:Still Renditions: thumbnail, screen.....	34
5.7.3	mpv:Still Renditions: alt.....	35
5.7.4	mpv:Still Derivations.....	35
5.8	Practices for MPV Basic Spec Usage.....	35
5.9	Practices for MPV Presentation Spec Usage	36
5.9.1	Attribute mpv:manifestLinkIdRef NOT USED.....	36
5.9.2	Only one <mpvp:Album> element.....	36
5.9.3	MUST NOT: Use <mpvp:AlbumRef> to link to another Album.....	36
5.10	Practices for MPV Music Profile Spec Usage	36
5.11	Practices for MPV Extensions	36
Chapter 6:	MPV CE Playlist Playback Practices.....	37
Appendix I:	References	38

Table of MPV CE Playlists Requirements

The following table is a comprehensive list of the requirements specified by the MPV CE Playlists specification, ordered by Requirement Number. This list can provide a quick reference guide to the specification and also is useful to quickly lookup more information about a specific requirement by number.

WRITER REQUIREMENTS

PLW100-1 mpvd:writtenBy attribute. The file:Manifest element MUST have the mpvd:writtenBy attribute, which identifies the application that wrote this specific MPV manifest. The attribute is updated everytime the manifest is modified in any way.	17
PLW100-2 mpvd:documentIdRef attribute. The file:Manifest element SHOULD have the mpvd:documentIdRef attribute and MUST have the attribute if the manifest is derived from another manifest.....	17
PLW100-3 Locale of the Manifest. The mpv:Document asset representing the manifest itself SHOULD have the dc:language element set to the default locale of the manifest's content.....	18
PLW100-4 MPV Document Profile. The MPV Document profile MUST be implemented and declared.....	18
PLW100-5 dc plain-text elements: creator, description, identifier, publisher, rights, title. When information corresponding to any or all of this basic set of defined Dublin Core elements is known, the appropriate elements SHOULD be used to represent the information. If present, an element's value MUST be UTF8-encoded plain text and has no specific interpretation other than as human-readable plain text that may be displayed or searched. An asset may have only a single occurrence of a given dc element, and only the singleton plain-text elements may be used.....	21
PLW100-6 dc:format element. Each asset that has at least one mpv:LastURL element MUST provide the file format of the asset's file content using the dc:format element as recommended by [MPVCore].	22
PLW100-7 dc:language element. dc:language SHOULD be used when asset language is known. If present, dc:language MUST be interpreted as a locale as recommended in [DC-NMF].....	22
PLW100-8 dc:subject element. dc:subject SHOULD be used when keywords about the asset subject are known. If present, dc:subject MUST be interpreted as one or more UTF8-encoded plain text keywords or keyphrases that use the semicolon (;) without spaces as a separator. The semicolon may be used in a keyword or keyphrase by escaping it with a backslash (;). A backslash may be used by escaping it with a backslash (\\). This interpretation is a refinement of the usage defined in [MPVCore].	23
PLW100-9 dcterms:created and modified elements. dcterms:created and dcterms:modified SHOULD be used when these dates are known.. If present, the value MUST be encoded as specified in [MPVCore].	23
PLW100-10 Screen rendition. Every visual asset SHOULD have a screen rendition and non-visual assets MAY have a screen rendition. Screen renditions MUST be visual assets, e.g. mpv:Still, mpv:StillMultishotSequence, mpv:StillPanoramaSequence, mpv:Video, and the resolution MUST be at least corresponding to typical screen resolution among current products, such as VGA or 1MPixel 2D resolution.....	30

PLW100-11 Thumbnail rendition. Every visual asset MUST have a thumbnail rendition and non-visual assets MAY have a thumbnail rendition. Thumbnail renditions MUST be simple visual assets, e.g. mpv:Still or mpv:Video, and the resolution MUST be less than VGA resolution. 30

PLW100-12 Show rendition. Every composite asset MAY have a Show rendition and non-visual assets MAY have a thumbnail rendition. Thumbnail renditions MUST be simple visual assets, e.g. mpv:Still or mpv:Video, and the resolution MUST be less than VGA resolution. 31

PLW100-13 mpv:LastURL mpv:filesystem attribute. Each asset that has at least one mpv:LastURL element must provide at least one mpv:LastURL element for each filesystem of the storage media on which the MPV playlist is resident and with which the asset can be accessed. 32

PLW100-14 mpv:LastURL UTF-8 and URL-encoded values. The value of a mpv:LastURL element MUST be UTF-8-encoded and URL-encoded..... 33

PLW100-15 mpv:Still Image size metadata. Every still image MUST have the image size metadata..... 33

READER REQUIREMENTS

PLR100-1 Filesystem-based Asset Access. The appropriate storage media filesystem MUST be used to access asset files referenced by MPV assets. Direct addressing to fixed locations on the storage media is not supported. 16

PLR100-2 dc metadata use. If a player supports display of Dublin Core-type information, it MUST support display of dc content in a MPV manifest, regardless of whether the reader can read the asset file format. dc values MUST take precedence over any equivalent values embedded in the asset itself. 23

PLR100-3 Screen rendition. A reader MUST display screen renditions of assets when the reader device or application can do visual display at screen resolutions. When they are present, screen renditions MUST be displayed for the following asset types: mpv:Still, mpv:Video, mpv:Audio, mpv:ManifestLink. 30

PLR100-4 Thumbnail rendition. A reader MUST display thumbnail renditions of assets when the reader device or application can do visual display at thumbnail resolutions. When they are present, thumbnail renditions MUST be displayed for the following asset types: mpv:Still, mpv:Video, mpv:ManifestLink. 31

PLR100-5 Show rendition. A reader SHOULD display the Show rendition of assets when the reader device or application can do visual display at thumbnail resolutions. Thumbnail renditions SHOULD be displayed for the following asset types: mpv:Still, mpv:Video, mpv:ManifestLink..... 31

PLR100-6 mpv:Still Renditions: alt. Readers SHOULD first try to use the primary asset referenced by the mpv:LastURL value. However, when the asset cannot be found or the asset file format cannot be understood, the reader SHOULD use one of the alt Renditions, if present..... 31

PLR100-7 mpv:LastURL Fragment Identifier Processing. Fragment identifiers in mpv:LastURL values MUST be allowed and handled gracefully, even if the specific fragment identifier understood. 33

PLR100-8 mpv:Still Renditions: thumbnail, screen. When present, readers SHOULD use thumbnail and screen renditions of mpv:Still when displaying thumbnail and screen resolution images because they minimize memory usage and enhance performance. 34

PLR100-9 mpv:Still Renditions: alt. Readers SHOULD first try to use the primary asset referenced by the mpv:LastURL value. However, when the asset cannot be found or the asset file format cannot be understood, the reader SHOULD use one of the alt Renditions, if present..... 35

PLR100-10 mpv:Still Renditions. Readers must support two types of mpv:renditionUsage for mpv:Still: thumbnail, screen 35

COMMON REQUIREMENTS

PLC100-1 Top-level mpv:Audio. A mpv:Audio in the mpv:AssetList that has is not referenced by any other asset. 24

PLC100-2 Primary mpv:Audio. A mpv:Audio in the mpv:AssetList that is referenced by a mpv:StillWithAudio, mpv:Seq, mpv:Par.	24
PLC100-3 Top-level mpv:Still. A mpv:Still in the mpv:AssetList that has is not referenced by any other asset.	25
PLC100-4 Primary mpv:Still. A mpv:Still in the mpv:AssetList that is referenced by a mpv:StillWithAudio, mpv:StillMultishotSequence, mpv:StillPanoramaSequence, or mpv:Seq.	25
PLC100-5 Thumbnail Rendition mpv:Still. A mpv:Still in the mpv:AssetList that is referenced by any asset using mpv:Rendition mpv:renditionUsage="thumbnail".	26
PLC100-6 Screen Rendition mpv:Still. A mpv:Still in the mpv:AssetList that is referenced by any asset using mpv:Rendition mpv:renditionUsage="screen".	26
PLC100-7 Artwork Related mpv:Still. A mpv:Still in the mpv:AssetList that is referenced by any asset using mpv:Related mpv:renditionUsage="urn:osta-org:mpv:music:artwork".	26
PLC100-8 Top-level mpv:StillWithAudio. A mpv:StillWithAudio in the mpv:AssetList that has is not referenced by any other asset.	27
PLC100-9 Top-level mpv:StillMultishotSequence. A mpv:StillMultishotSequence in the mpv:AssetList that has is not referenced by any other asset.	27
PLC100-10 Alt Rendition mpv:StillMultishotSequence. A mpv:StillMultishotSequence in the mpv:AssetList that is referenced by a mpv:Video asset.	27
PLC100-11 Top-level mpv:StillPanoramaSequence. A mpv:StillPanoramaSequence in the mpv:AssetList that has is not referenced by any other asset.	28
PLC100-12 Top-level mpv:Video. A mpv:Video in the mpv:AssetList that has is not referenced by any other asset.	29
PLC100-13 Show Rendition mpv:Video. A mpv:Video in the mpv:AssetList that is referenced as a mpv:Rendition mpv:renditionUsage="show" by the following composite asset: mpvp:Album, mpv:StillMultishotSequence, mpv:StillPanoramaSequence, mpv:StillWithAudio, mpv:Seq, mpv:Par.	29
PLC100-14 Asset Relationships. Five categories of asset relationships are highlighted for interoperability: Master asset; alt, thumbnail, screen, and subsampled Rendition assets; derivedFrom Related assets. Both Writers and Readers SHOULD support these relationships.	29
PLC100-15 Master asset. Every MPV asset MUST have a master asset, which is a root asset in the asset list or in the mpvp:Album playlist and is the asset that conceptually is the center of the user's interaction. In other words, the Master asset corresponds most naturally with the user's conceptual model.	30
PLC100-16 Attribute mpv:manifestLinkIdRef. The attribute mpv:manifestLinkIdRef MUST NOT be used on any element.	36
PLC100-17 Only one <mpvp:Album>. Zero or one <mpvp:Album> elements MUST be present, but not more.	36
PLC100-18 <mpvp:AlbumRef> NOT USED. The <mpvp:AlbumRef> MUST NOT be used.	36

Chapter 1: Introduction

1.1 Executive Summary

MPV (MusicPhotoVideo) is an open, multiplatform specification for playlists and asset management of digital music, photo, and video files. MPV makes easier the representation, exchange, processing and playback of collections of digital media content, including music, still images, stills with audio, still sequences, video clips, and audio clips.

MPV CE Playlists Profile, which is defined by this specification, is a well-defined usage of MPV that provides guaranteed interoperability between playlist creation and playlist reader applications and devices.

MPV CE Playlists can be enjoyed in consumer electronics products such as a CD or DVD player or on a PC. MPV CE Playlists and the content they reference are unlike audio CDs and DVD-Video discs because they store the multimedia content like MP3 and JPEG files in formats used by PCs on a “data CD”.

MPV is an open format developed under the leadership of the Optical Storage Technology Association (OSTA) and available from OSTA at no cost. Information regarding the MPV Specification, SDK for software developers, and logo licensing program can be found at www.osta.org/mpv.

The MPV Logo and sublogos provide a consumer-oriented mark of compatibility that can be used by creative applications and devices, on storage media, and by playback devices and applications. Use of the MPV Logo and sublogos is governed by a licensing program that defines the requirements and costs.

SITUATION TODAY

Consumers create CDs full of personal digital content – music, photo, and video content in PC file formats like MP3, JPEG and MPEG – captured by digital cameras, photofinishing retailers, personal music collections, and PC multimedia applications. The expectation and desire of consumers is to enjoy the playback experience not only in PCs but wherever a player can go -- in their home entertainment environment, in their car or in their pocket.

Today, most consumer electronic (CE) devices do not recognize the content on CDs created by PC applications or retail services or do so poorly. Because each application stores the content on a disc uniquely, there is no standard way for CD and DVD players to recognize and playback the content. And the user playback experience is different between each CE device.

Without a standard method for organization and access, the CE device can take many minutes reading through large collections of multimedia content or will present filesystem views of the data. Consumers get frustrated with trying to find and quickly access their desired music, photo, or video content.

CE devices are starting to add support for PC formats. MP3 for music has enjoyed wide spread adoption in DVD player, car stereo, and personal music systems. Support for JPEG for photos is growing in consumer electronics

products. MPEG1 and MPEG2 video is already broadly adopted by both PC and CE industries. Additional formats are emerging and growing in popularity, such as Windows Media Audio (WMA), ATRAC3, MPEG4 Audio (AAC), and more.

MPV BENEFITS CONSUMERS

When applications and application both write and read MPV, consumers will enjoy enhanced interoperability of content between applications and devices from any vendor. This gives consumers more choice and vendors greater ability to innovate and differentiate.

The MPV CE Playlists specification provides interoperability of navigation and asset management. Between products that produce compatible formats, MPV CE Playlists provide an excellent interoperability experience.

For consumers burning CDs on PCs, applications supporting MPV will create MPV CDs that can be viewed in other PCs and provide the additional benefit of playing back in CE devices such as CD or DVD players from many manufacturers. For consumers sending film or digital content to a retail photo service for storage on a photo CD, MPV CDs will provide an archive of the high resolution content as well as a consistent playback experience in DVD players and TV screens.

If digital cameras implement MPV, consumers wishing to enjoy the playback experience in their TV or DVD system will benefit from MPV organization to maintain the user playback experience when the memory card is removed from the camera. TVs or DVDs supporting MPV will playback the content from any camera in a consistent manner.

1.2 MPV CE Playlists – Guaranteed Navigation, Enhanced Playback Experience

MPV CE Playlists provide a standard way to organize and navigate content and to play it back. The MPV CE Playlists control file – think of it as a table of contents or index – on the disc or flash memory card which a CE device with a MPV reader can recognize and play. The control file defines the contents of the storage media and enables the CE device to quickly find, access and present the consumer's multimedia content. MPV CE Playlist-enabled playback devices quickly recognize the control file and present the user with a playlist and simple navigation of the content. MPV CE Playlists allow the user or creative application to organize the media files any way they wish on the storage media and using long user-friendly filenames. Additions and changes to playlists can be made easily and quickly without rewriting the entire storage media.

In addition to providing basic playlist capability of a sequence of files, MPV CE Playlists can enhance the user experience in significant ways, such as:

- Users can navigate large collections of hundreds or thousands of files organized into multiple playlists, such as songs by genre, by artist, and by album.
- Creation and playback of a multimedia slideshows of pictures and video with background music and transition effects.
- Songs can have album art, lyrics, music in both audio-only and video file formats, and multiple encodings of the same music, for example WMA and MP3.
- MPV manages multiple renditions of images, such as high resolution, screen resolution, and thumbnails that enable high-performance playback on low-end systems using low-res images and printing using high-res images.

This document establishes a fixed usage of MPV that is used to guarantee interoperability between playlist creator and playlist reader. However, it does not require any specific set of media file formats. Thus, it is quite possible to have a storage media filled with media files and MPV CE Playlists that a MPV-enabled player can navigate but cannot play because it cannot read the media file formats. While this can be a disappointing user experience, there is still tremendous value in establishing an interoperable playlists format because it can be stable while media file

formats continue to evolve and because many users are willing and able to ensure that the media file formats they use and their playback devices are compatible.

1.3 No Content Playback Guarantee

The MPV CE Playlist specification does NOT establish a required set of media file content formats that must be produced by writer applications and must be played by reader applications and devices. While the MPV CE Playlist specification ensures that a MPV player can properly read the MPV playlist information and present its contents to the user, it does NOT ensure that the content itself can be played by the MPV player.

To provide guarantees of content playback, the MPV CE Playlists specification can be combined with a well-defined set of content file formats. The ability to do this is available to any entity or organization that wants to do so. In a typical content types specification, the specification would reference the MPV CE Playlists specification and provide a well-defined set of content formats to be used in combination.

The content formats would be of types relevant to the entity or organization. For example, an audio books standard might specify only audio formats, while a digital camera standard might specify audio, photo, and video-clip formats.

1.4 MPV Logo Program

The MPV Logo communicates compatibility to consumers. A license from OSTA is required to use the MPV Logo. License terms include satisfactory demonstrated compliance with the logo license requirements. A modest license fee is required to cover administrative costs; however, OSTA does not charge a royalty for use of the specification or logo. A compliance test suite and procedures and compliance testing materials are made available and administered by OSTA.

OSTA will manage the MPV logo licensing program for software application developers as well as CE device manufacturers. Conformance test suites will be self-administered. Where applicable, conformance and support of the defined file formats will provide users with a guarantee of capability. For more information on the logo licensing program, see www.osta.org/mpv.

This MPV logo establishes a fixed usage of MPV that is used to guarantee interoperability between playlist creator and playlist reader. However, it does not require any specific set of media file formats.

1.5 MPV Profiles

The MPV specification is developed in a modular manner and in phases. This results in "profiles". Each profile in MPV defines only those formats and practices that are necessary for the key tasks targeted by the profile. MPV CE Playlists Profile makes use of several other profiles, including:

- **Basic Profile:** key tasks: defining content collections, renditions, identifiers, and access to other metadata
- **Presentation Profile:** two key tasks: viewing a slideshow and interactively browsing content collections
- **Music Profile:** key tasks: listening to a music collection and interactively browsing content collections
- **Document Profile:** key task: identifying MPV manifest documents and their origin

Underlying all profiles is the “Core”, which defines the overall framework of all MPV profiles. The Basic and Presentation Profiles, for example, build on the Core and, when implemented in consumer electronics devices like DVD players or in application software, can provide compelling playback of photo-video slideshows and interactive browsing of photo-video content. It can also facilitate interchange of photo-video content between applications. The Presentation Profile is also used by the Music Profile to as a music playlist.

1.6 Terms of Use

This section of the specification is descriptive and not intended to be complete nor definitive. Please refer to the definitive statement of licensing terms at the beginning of the MPV specification document for a precise and legal description.

The MPV specification is developed using an open process. The resulting specification is available from OSTA. No royalty is charged by OSTA for use of the specification. The overall desire is to develop a specification that is not subject to separate licensing requirements or royalty. During the development process, the expectation is that all participants contribute their efforts and intellectual property without any expectation or requirement for compensation. However, OSTA does not warrant that the specification is not or will not be subject to such claims by other parties.

Chapter 2: MPV CE Playlists Profile 1.0

2.1 MPV CE Playlists Introduction

The MPV CE Playlists Profile makes use of existing MPV specifications ([MPVCore], [MPVBasic], [MPVPres], [MPVMusic], [MPVDocument]) and combines them with additional specific requirements to define tightly the usage of these MPV profiles to guarantee interoperability between devices and applications that conform to the MPV CE Playlists specification.

Conformance with the MPV CE Playlists specification is required for use of the MPV Logo on products.

The MPV CE Playlists Profile introduces no new schema or metadata. The entire MPV CE Playlists specification consists of practices that dictate how other the MPV specifications are used. The objective of the MPV CE Playlists specification is to make consistent both MPV playlist writing, reading, and playback. Notably, this is achieved in some cases by overriding or limiting the use of certain aspects of the referenced MPV specifications, where their capabilities are too broad, to imprecise, or otherwise unsuitable for broad implementation across a full range of applications and devices, including those with limited capacities, such as limited RAM and FlashROM memory and CPU and I/O performance.

These limitations and consistent practices enable writers and readers to be highly interoperable, enabling consumers to expect and experience with compatibility and interoperability of MPV CE Playlists.

2.2 MPV CE Playlists Specification Practices

The MPV CE Playlists Specification establishes three important practices.

1. All practices are qualified using the keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, if and where they appear in this document, are to be interpreted as described in [RFC2119].

The keywords are classified into three imperative levels. All words at a given level have the same level of imperative.

- Level 1: MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT
- Level 2: SHOULD, SHOULD NOT
- Level 3: RECOMMENDED, MAY, OPTIONAL

For conformance testing, the keyword imperative levels are treated as warning levels, with the following meaning:

- Level 1: Error -- MUST be fixed.
 - Level 2: Severe Warning -- SHOULD be fixed to enhance compatibility.
 - Level 3: Warning – RECOMMEND to be fixed. Not critical to compatibility.
2. All practices are classified as either Common, Writer, or Reader requirements. Common requirements are practices that are relevant to both Writers and Readers. Writer requirements identify specific MPV content that Writers must produce; Reader requirements identify specific behaviours that Readers must implement.
 3. All practices are identified with a unique “requirement number” by which they may be referenced easily.

For example, a verification tools could output results and reference the requirement by number. Practices are identified using a prefix plus a number. This specification uses the letters “PL” plus the letter “C” for common, “W” for writer, or “R” for reader, plus the specification version number without the decimal separator as the prefix, as in PL100-83. A future revision of the specification might identify the same requirement by a different number if the number or order of the requirements were to change, e.g. PL110-85.

2.3 Formalities For Use of the MPV CE Playlists Profile

The mechanism that MPV uses to add capabilities to the Core specification is the Profile. MPV Core sets out specific formalities to follow when a MPV Profile is used -- an MPV file must declare which profiles it implements and it must declare the namespaces of the profiles. This allows a processing application to quickly determine whether a given MPV file meets its expectations for processing.

PROFILE COMPONENTS

The MPV CE Playlists Profile 1.0 makes use of four other specifications:

- MPV Core 1.0 Specification
- MPV Presentation Profile 1.0 Specification
- MPV Music Profile 1.0 Specification
- MPV Documents Profile 1.0 Specification

The MPV CE Playlists Profile 1.0 includes the schema and practices detailed by this document.

COMPATIBILITY

The MPV CE Playlists Profile 1.0 is fully compatible with the MPV framework established by [MPVCore]. Thus MPV files that implement the MPV CE Playlists Profile 1.0 should be usable in basic ways by most MPV-aware applications and devices that are not in conformance with the MPV CE Playlists Profile.

Applications and devices that conform to the MPV CE Playlists 1.0 specification will provide the user a consistent user experience.

SCHEMA NAMESPACE

The MPV CE Playlists specification does not define any schema. Therefore no schema namespace is defined.

PROFILE IDENTIFIER

This information must be present in the Profile section of the MPV Manifest.

MPV CE Playlists Profile Name	http://ns.osta.org/mpv/playlists/1.0/
----------------------------------	---

EXAMPLE

Chapter 3: MPV CE Playlists – Filesystem Requirements

3.1.1 Filesystem-based Asset Access

*PLR100-1 Filesystem-based Asset Access. The appropriate storage media filesystem **MUST** be used to access asset files referenced by MPV assets. Direct addressing to fixed locations on the storage media is not supported.*

Chapter 4: MPV CE Playlists – Manifest Spec Practices

4.1 Practices for MPV Document spec

4.1.1 mpvd:writtenBy attribute

PLW100-1 mpvd:writtenBy attribute. The file:Manifest element MUST have the mpvd:writtenBy attribute, which identifies the application that wrote this specific MPV manifest. The attribute is updated everytime the manifest is modified in any way.

This practice conforms to [MPVDoc] and is to guarantee that a reader can quickly determine whether it knows the application that wrote the manifest. Because this attribute is in the opening element of the manifest, it enables easy transfer of the manifest processing to one of several possible readers that provide differing behaviours and levels of validation. This attribute can also be used by a MPV launcher to preferentially launch the application that wrote the manifest. Note that the value of the attribute MUST be a URN-style value to minimize the risk of value collision.

Example:

```
<file:Manifest
  mpvd:writtenBy="http://www.mycompany.com/myapp/1.05/"
  mpvd:documentIdRef="ID000001"
  xmlns:file="http://ns.osta.org/manifest/1.0/"
  xmlns:mpv="http://ns.osta.org/mpv/1.0/"
  xmlns:mpvp="http://ns.osta.org/mpv/presentation/1.0/"
  xmlns:mpvm="http://ns.osta.org/mpv/music/1.0/"
  xmlns:mpvd="http://ns.osta.org/mpv/document/1.0/"
  xmlns:nmf="http://ns.osta.org/nmf/1.0/" >
  ...
</file:Manifest>
```

4.1.2 mpvd:documentIdRef attribute

PLW100-2 mpvd:documentIdRef attribute. The file:Manifest element SHOULD have the mpvd:documentIdRef attribute and MUST have the attribute if the manifest is derived from another manifest.

This practice conforms to [MPVDoc] and is very useful for applications that are looking for the rich metadata information about the current document. For example, a typical use will be to look up the referenced mpv:Document asset, then look at its <mpv:Related mpv:relationship="derivedFrom"> element to locate a previous manifest from which the current manifest was derived.

Example:

```
<file:Manifest
  mpvd:writtenBy="http://www.mycompany.com/myapp/1.05/"
  mpvd:documentIdRef="ID000001"
  xmlns:file="http://ns.osta.org/manifest/1.0/"
  xmlns:mpv="http://ns.osta.org/mpv/1.0/"
  xmlns:mpvp="http://ns.osta.org/mpv/presentation/1.0/"
  xmlns:mpvm="http://ns.osta.org/mpv/music/1.0/"
  xmlns:mpvd="http://ns.osta.org/mpv/document/1.0/"
  xmlns:nmf="http://ns.osta.org/nmf/1.0/" >
...
</file:Manifest>
```

4.1.3 Locale of the Manifest

PLW100-3 Locale of the Manifest. The mpv:Document asset representing the manifest itself SHOULD have the dc:language element set to the default locale of the manifest's content.

This practice conforms to [MPVDoc]. The MPV manifest is authored for a single “locale”, where a locale is a combination of language and territory, such as English-U.S. The locale of a manifest is recorded as a dc:language element of the manifest's own mpv:Document asset. The default locale of all text in the manifest is identified by the dc:language element within the mpv:Document element representing the current manifest. A player MAY honor the use of dc:language for purposes of sorting, line wrapping, currency, and other values.

Example:

```
<file:Manifest
  mpvd:writtenBy="http://www.mycompany.com/myapp/1.05/"
  mpvd:documentIdRef="ID000001"
  xmlns:file="http://ns.osta.org/manifest/1.0/"
  xmlns:mpv="http://ns.osta.org/mpv/1.0/"
  xmlns:mpvp="http://ns.osta.org/mpv/presentation/1.0/"
  xmlns:mpvm="http://ns.osta.org/mpv/music/1.0/"
  xmlns:mpvd="http://ns.osta.org/mpv/document/1.0/"
  xmlns:nmf="http://ns.osta.org/nmf/1.0/" >
...
  <mpv:AssetList>
    <!-- This asset describes this very MPV Manifest -->
    <mpv:Document mpv:id="ID000001">
      <mpv:InstanceID>urn.osta-org.mpv.uuid.2234BDF9BBA934338DFBFFDE8342</mpv:InstanceID>
      <nmf:Metadata>
        <Properties xmlns="http://purl.org/dc/elements/1.1/"
...
          <language>en-US</language>
...
        </Properties>
      </nmf:Metadata>
    </mpv:Document>
  </file:Manifest>
```

4.1.4 MPV Document Profile

PLW100-4 MPV Document Profile. The MPV Document profile MUST be implemented and declared.

This practice confirms that the manifest implements the MPV Document profile.

Example:

```
<file:Manifest
  mpvd:writtenBy="http://www.mycompany.com/myapp/1.05/"
  mpvd:documentIdRef="ID000001"
  xmlns:file="http://ns.osta.org/manifest/1.0/"
  xmlns:mpv="http://ns.osta.org/mpv/1.0/"
  xmlns:mpvp="http://ns.osta.org/mpv/presentation/1.0/"
  xmlns:mpvm="http://ns.osta.org/mpv/music/1.0/"
  xmlns:mpvd="http://ns.osta.org/mpv/document/1.0/"
  xmlns:nmf="http://ns.osta.org/nmf/1.0/" >
<nmf:Metadata>
  <ManifestProperties xmlns="http://ns.osta.org/manifest/1.0/">
    <ProfileBag>
      <Profile>http://ns.osta.org/mpv/basic/1.0/</Profile>
      <Profile>http://ns.osta.org/mpv/presentation/1.0/</Profile>
      <Profile>http://ns.osta.org/mpv/music/1.0/</Profile>
      <Profile>http://ns.osta.org/mpv/document/1.0/</Profile>
      <Profile>http://ns.osta.org/mpv/playlists/1.0/</Profile>
    </ProfileBag>
  </ManifestProperties>
</nmf:Metadata>
...
</file:Manifest>
```

4.2 Example

This is an example of the [MPVDoc]-conformant parts of a well-formed MPV CE Playlist manifest.

```
<?xml version="1.0" encoding="UTF-8"?>
<file:Manifest
  mpvd:writtenBy="http://www.mycompany.com/myapp/1.05/"
  mpvd:documentIdRef="ID000001"
  xmlns:file="http://ns.osta.org/manifest/1.0/"
  xmlns:mpv="http://ns.osta.org/mpv/1.0/"
  xmlns:mpvp="http://ns.osta.org/mpv/presentation/1.0/"
  xmlns:mpvm="http://ns.osta.org/mpv/music/1.0/"
  xmlns:mpvd="http://ns.osta.org/mpv/document/1.0/"
  xmlns:nmf="http://ns.osta.org/nmf/1.0/"
  xml:lang="en-US" >
<nmf:Metadata>
  <ManifestProperties xmlns="http://ns.osta.org/manifest/1.0/">
    <ProfileBag>
      <Profile>http://ns.osta.org/mpv/basic/1.0/</Profile>
      <Profile>http://ns.osta.org/mpv/presentation/1.0/</Profile>
      <Profile>http://ns.osta.org/mpv/music/1.0/</Profile>
      <Profile>http://ns.osta.org/mpv/document/1.0/</Profile>
      <Profile>http://ns.osta.org/mpv/playlists/1.0/</Profile>
    </ProfileBag>
  </ManifestProperties>
</nmf:Metadata>
```

```
<mpv:AssetList>
  <!-- This asset describes this very MPV Manifest -->
  <mpv:Document mpv:id="ID000001">
    <mpv:InstanceID>urn.osta-org.mpv.uuid.2234BDF9BBA934338DFBFFDE8342</mpv:InstanceID>
    <nmf:Metadata>
      <Properties xmlns="http://purl.org/dc/elements/1.1/"
        <creator>MyCompany MyApp 1.0</creator>
        <description>this is a description</description>
        <language>en-US</language>
        <title>some title</title>
      </Properties>
      <Properties xmlns="http://purl.org/dc/terms/1.0/"
        <createDate></createDate>
        <modifiedDate>asdf </modifiedDate>
      </Properties>
    </nmf:Metadata>
    <mpv:Related mpv:relationship="derivedFrom">
      <mpv:DocumentRef mpv:idRef="ID000002"/>
    </mpv:Related>
  </mpv:Document>
  <!-- This asset describes the MPV Manifest from which the current one was derived -->
  <mpv:Document mpv:id="ID000002">
    <mpv:InstanceID>urn.osta-org.mpv.uuid.4583BDF9BBA934338DFBFF993493</mpv:InstanceID>
    <mpv:LastURL mpv:filesystem="NTFS">album002.pvm</mpv:LastURL>
  </mpv:Document>
  ...
</mpv:AssetList>

</file:Manifest>
```

Chapter 5: MPV CE Playlists – Core Spec Practices

5.1 Practices for Asset Dublin Core metadata

5.1.1 dc plain-text elements: creator, description, identifier, publisher, rights, title

PLW100-5 dc plain-text elements: creator, description, identifier, publisher, rights, title. When information corresponding to any or all of this basic set of defined Dublin Core elements is known, the appropriate elements SHOULD be used to represent the information. If present, an element's value MUST be UTF8-encoded plain text and has no specific interpretation other than as human-readable plain text that may be displayed or searched. An asset may have only a single occurrence of a given dc element, and only the singleton plain-text elements may be used.

This practice is to guarantee that a reader can read basic Dublin Core information regardless of its ability to read the information from the file format. Also, by caching the value in the MPV manifest, it protects that value from loss if the asset itself should be edited. The element value is arbitrary with no specified interpretation other than as a human-readable plain-text string suitable for display or search.

Many applications may also have formatted versions of this content. MPV does not define the interchange of formatted content. The recommended approach is to store the formatted content using metadata extensions while recording the interchangeable form of the information as DC metadata.

EXAMPLE:

```
<mpv:Still mpv:id="ID000103">
  <mpv:ContentID>urn:osta-org:mpv:dsig:md5:all:EF886AEFA3B340da971BAF09B17DBC12</mpv:ContentID>
  <mpv:LastURL>IMG001.JPG</mpv:LastURL>
  <nmf:Metadata>
    <!-- DC provides baseline interchange of metadata. -->
    <dc:Properties>
      <dc:description>Formatted content that may have any kind of rich text and even be
      spread across multiple metadata elements will be provided as plain-text in the DC
      element.</dc:description>
      <dc:format>image/jpeg</dc:format>
      <dc:title>Title line 1 Title line 2 Title line 3</dc:title>
    </dc:Properties>
    <!-- the app defines its own metadata schema to hold formatted content. -->
```

```

<app:Properties>
  <app:description><p><i>Formatted</i> content that may have <b>any</b> kind of rich
text and even be spread across multiple <a href="glossary.htm#metadata">metadata elements</a>
will be provided as <i>plain-text</i> in the DC element.</p></app:description>
  <app:title><font size=16><p><b>Title line 1</b></p></font><font size=14><p><i>Title
line 2</i></p></font><font size=12><p>Title line 3</p></font></app:title>
</app:Properties>
</nmf:Metadata>
</mpv:Still>

```

5.1.2 dc:format element

PLW100-6 dc:format element. *Each asset that has at least one mpv:LastURL element MUST provide the file format of the asset's file content using the dc:format element as recommended by [MPVCore].*

This practice is to guarantee that a reader can quickly determine whether it can handle the asset file format without opening up the asset file. On devices with slow I/O, this is an importance performance enhancement.

[MPVCore] provides a set of well-defined MIME type values for the dc:format element in the [MPVCore] specification. This specification may be updated over time to add additional values as needed.

EXAMPLE:

```

<mpv:Still mpv:id="ID000103">
  <mpv:ContentID>urn:osta-org:mpv:dsig:md5:all:EF886AEFA3B340da971BAF09B17DBC12</mpv:ContentID>
  <mpv:LastURL>IMG001.JPG</mpv:LastURL>
  <nmf:Metadata>
    <dc:Properties>
      <dc:format>image/jpeg</dc:format>
    </dc:Properties>
  </nmf:Metadata>
  <mpv:Rendition>
    <mpv:StillRef mpv:idRef="ID000103T"/>
  </mpv:Rendition>
</mpv:Still>
<mpv:Still mpv:id="ID000103T">
  <mpv:ContentID>urn:osta-org:mpv:dsig:md5:all:EF886AEFA3B340da971BAF09B17DBC12</mpv:ContentID>
  <mpv:LastURL>IMG001.JPG#vnd.osta-org.exif-thumb</mpv:LastURL>
  <nmf:Metadata>
    <dc:Properties>
      <dc:format>image/vnd.osta-org.exif-thumb</dc:format>
    </dc:Properties>
  </nmf:Metadata>
</mpv:Still>

```

5.1.3 dc:language element

PLW100-7 dc:language element. *dc:language SHOULD be used when asset language is known. If present, dc:language MUST be interpreted as a locale as recommended in [DC-NMF].*

This practice is to guarantee that a reader can determine the locale of the information regardless of its ability to read the information from the file format.

5.1.4 dc:subject element

PLW100-8 dc:subject element. dc:subject SHOULD be used when keywords about the asset subject are known. If present, dc:subject MUST be interpreted as one or more UTF8-encoded plain text keywords or keyphrases that use the semicolon (;) without spaces as a separator. The semicolon may be used in a keyword or keyphrase by escaping it with a backslash (\;). A backslash may be used by escaping it with a backslash (\\). This interpretation is a refinement of the usage defined in [MPVCore].

This practice is to guarantee that a reader can read keyword information regardless of its ability to read the information from the file format. Also, by caching the value in the MPV manifest, it protects that value from loss if the asset itself should be edited. Each keyword or phrase is an arbitrary value with no specified interpretation other than as a human-readable plain-text string.

5.1.5 dcterms:created and :modified elements

PLW100-9 dcterms:created and modified elements. dcterms:created and dcterms:modified SHOULD be used when these dates are known.. If present, the value MUST be encoded as specified in [MPVCore].

This practice is to allow a reader can display date information without having to access the filesystem. For internet usage, this is especially valuable.

5.1.6 dc metadata use

PLR100-2 dc metadata use. If a player supports display of Dublin Core-type information, it MUST support display of dc content in a MPV manifest, regardless of whether the reader can read the asset file format. dc values MUST take precedence over any equivalent values embedded in the asset itself.

This practice is to encourage that a reader can read Dublin Core information regardless of its ability to read the information from the file format. This makes the playback experience more robust because when the reader cannot read an asset's file format, it can at least display information about the asset.

If a player supports display of basic Dublin Core information, it MUST support the following essential Dublin Core metadata.

DC Elements

- creator
- description
- format
- identifier
- language
- publisher
- rights
- subject
- title

DC Terms

- created
- modified

5.2 Well-Known Music Asset Types

The mpv:AssetList can contain many assets organized in a rich variety of ways. For purposes of MPV CE Playlists, the following Photo Asset Types are defined as “well known” constructs that must be supported.

5.2.1 Top-level mpv:Audio

PLC100-1 Top-level mpv:Audio. A mpv:Audio in the mpv:AssetList that has is not referenced by any other asset.

EXAMPLE:

```
<mpv:AssetList>
  <mpv:Audio mpv:id="ID000101">
  ...
  </mpv:Audio>
</mpv:AssetList>
```

5.2.2 Primary mpv:Audio

PLC100-2 Primary mpv:Audio. A mpv:Audio in the mpv:AssetList that is referenced by a mpv:StillWithAudio, mpv:Seq, mpv:Par.

EXAMPLE:

```
<mpv:AssetList>
  <mpv:StillWithAudio mpv:id="ID000101">
  ...
    <mpv:StillRef mpv:idRef="ID000102"/>
  ...
  </mpv:StillWithAudio>
  <mpv:Still mpv:id="ID000102">
  ...
  </mpv:Still>
</mpv:AssetList>
```

5.3 Well-Known Photo Asset Types

The mpv:AssetList can contain many assets organized in a rich variety of ways. For purposes of MPV CE Playlists, the following Photo Asset Types are defined as “well known” constructs that must be supported.

5.3.1 Top-level mpv:Still

PLC100-3 Top-level mpv:Still. A mpv:Still in the mpv:AssetList that has is not referenced by any other asset.

EXAMPLE:

```
<mpv:AssetList>
  <mpv:Still mpv:id="ID000101">
  ...
  </mpv:Still>
</mpv:AssetList>
```

5.3.2 Primary mpv:Still

PLC100-4 Primary mpv:Still. A mpv:Still in the mpv:AssetList that is referenced by a mpv:StillWithAudio, mpv:StillMultishotSequence, mpv:StillPanoramaSequence, or mpv:Seq.

EXAMPLE:

```
<mpv:AssetList>
  <mpv:StillWithAudio mpv:id="ID000101">
  ...
  <mpv:StillRef mpv:idRef="ID000102"/>
  ...
  </mpv:StillWithAudio>
  <mpv:Still mpv:id="ID000102">
  ...
  </mpv:Still>
</mpv:AssetList>
```

EXAMPLE:

```
<mpv:AssetList>
  <mpv:Seq mpv:id="ID000101">
  ...
  <mpv:StillRef mpv:idRef="ID000102"/>
  <mpv:StillRef mpv:idRef="ID000103"/>
  </mpv:Seq>
  <mpv:Still mpv:id="ID000102">
  ...
  </mpv:Still>
  <mpv:Still mpv:id="ID000103">
  ...
  </mpv:Still>
</mpv:AssetList>
```

5.3.3 Thumbnail Rendition mpv:Still

PLC100-5 Thumbnail Rendition mpv:Still. A *mpv:Still* in the *mpv:AssetList* that is referenced by any asset using *mpv:Rendition mpv:renditionUsage="thumbnail"*.

EXAMPLE:

```
<mpv:AssetList>
  <mpv:Still mpv:id="ID000101">
  ...
    <mpv:Rendition mpv:renditionUsage="thumbnail">
      <mpv:StillRef mpv:idRef="ID000102"/>
    </mpv:Rendition>
  </mpv:Still>
  <mpv:Still mpv:id="ID000102">
  ...
  </mpv:Still>
</mpv:AssetList>
```

5.3.4 Screen Rendition mpv:Still

PLC100-6 Screen Rendition mpv:Still. A *mpv:Still* in the *mpv:AssetList* that is referenced by any asset using *mpv:Rendition mpv:renditionUsage="screen"*.

Screen renditions can be on any asset type, but are especially likely for *mpv:Still* and *mpv:ManifestLink* assets.

EXAMPLE:

```
<mpv:AssetList>
  <mpv:Still mpv:id="ID000101">
  ...
    <mpv:Rendition mpv:renditionUsage="thumbnail">
      <mpv:StillRef mpv:idRef="ID000102"/>
    </mpv:Rendition>
  </mpv:Still>
  <mpv:Still mpv:id="ID000102">
  ...
  </mpv:Still>
</mpv:AssetList>
```

5.3.5 Artwork Related mpv:Still

PLC100-7 Artwork Related mpv:Still. A *mpv:Still* in the *mpv:AssetList* that is referenced by any asset using *mpv:Related mpv:relationship="urn:osta-org:mpv:music:artwork"*.

EXAMPLE:

```
<mpv:AssetList>
  <mpv:Audio mpv:id="ID000101">
  ...
    <mpv:Related mpv:relationship="urn:osta-org:mpv:music:artwork">
      <mpv:StillRef mpv:idRef="ID000102"/>
    </mpv:Related>
  </mpv:Audio>
  <mpv:Still mpv:id="ID000102">
```

```

...
</mpv:Still>
</mpv:AssetList>

```

5.3.6 Top-level mpv:StillWithAudio

PLC100-8 Top-level mpv:StillWithAudio. A mpv:StillWithAudio in the mpv:AssetList that has is not referenced by any other asset.

EXAMPLE:

```

<mpv:AssetList>
  <mpv:StillWithAudio mpv:id="ID000101">
    ...
    <mpv:StillRef mpv:idRef="ID000102"/>
    <mpv:AudioRef mpv:idRef="ID000103"/>
  </mpv:StillWithAudio>
  <mpv:Still mpv:id="ID000102">
    ...
  </mpv:Still>
  <mpv:Audio mpv:id="ID000103">
    ...
  </mpv:Audio>
</mpv:AssetList>

```

5.3.7 Top-level mpv:StillMultishotSequence

PLC100-9 Top-level mpv:StillMultishotSequence. A mpv:StillMultishotSequence in the mpv:AssetList that has is not referenced by any other asset.

EXAMPLE:

```

<mpv:AssetList>
  <mpv:StillMultishotSequence mpv:id="ID000101">
    ...
    <mpv:StillRef mpv:idRef="ID000102"/>
    <mpv:StillRef mpv:idRef="ID000103"/>
  </mpv:StillMultishotSequence>
  <mpv:Still mpv:id="ID000102">
    ...
  </mpv:Still>
  <mpv:Still mpv:id="ID000103">
    ...
  </mpv:Still>
</mpv:AssetList>

```

5.3.8 Alt Rendition mpv:StillMultishotSequence

PLC100-10 Alt Rendition mpv:StillMultishotSequence. A mpv:StillMultishotSequence in the mpv:AssetList that is referenced by a mpv:Video asset.

EXAMPLE:

```

<mpv:AssetList>
  <mpv:Video mpv:id="ID000100">
  ...
    <mpv:Rendition mpv:renditionUsage="alt">
      <mpv:StillMultishotSequenceRef mpv:idRef="ID000101"/>
    </mpv:Rendition>
  </mpv:Video>
  <mpv:StillMultishotSequence mpv:id="ID000101">
  ...
    <mpv:StillRef mpv:idRef="ID000102"/>
    <mpv:StillRef mpv:idRef="ID000103"/>
  </mpv:StillMultishotSequence>
  <mpv:Still mpv:id="ID000102">
  ...
  </mpv:Still>
  <mpv:Still mpv:id="ID000103">
  ...
  </mpv:Still>
</mpv:AssetList>

```

5.3.9 Top-level mpv:StillPanoramaSequence

PLC100-11 Top-level mpv:StillPanoramaSequence. A mpv:StillPanoramaSequence in the mpv:AssetList that has is not referenced by any other asset.

EXAMPLE:

```

<mpv:AssetList>
  <mpv:StillPanoramaSequence mpv:id="ID000101">
  ...
    <mpv:StillRef mpv:idRef="ID000102"/>
    <mpv:StillRef mpv:idRef="ID000103"/>
  </mpv:StillPanoramaSequence>
  <mpv:Still mpv:id="ID000102">
  ...
  </mpv:Still>
  <mpv:Still mpv:id="ID000103">
  ...
  </mpv:Still>
</mpv:AssetList>

```

5.4 Well-Known Video Asset Types

The mpv:AssetList can contain many assets organized in a rich variety of ways. For purposes of MPV CE Playlists, the following Video Asset Types are defined as “well known” constructs that must be supported.

5.4.1 Top-level mpv:Video

PLC100-12 Top-level mpv:Video. A *mpv:Video* in the *mpv:AssetList* that has is not referenced by any other asset.

EXAMPLE:

```
<mpv:AssetList>
  <mpv:Video mpv:id="ID000101">
  ...
  </mpv:Video>
</mpv:AssetList>
```

5.4.2 Show Rendition mpv:Video

PLC100-13 Show Rendition mpv:Video. A *mpv:Video* in the *mpv:AssetList* that is referenced as a *mpv:Rendition mpv:renditionUsage="show"* by the following composite asset: *mpvp:Album*, *mpv:StillMultishotSequence*, *mpv:StillPanoramaSequence*, *mpv:StillWithAudio*, *mpv:Seq*, *mpv:Par*.

Show rendition videos are a way for writer applications and devices to add value to MPV collections by creating high-value “shows” of composite assets. The Playlist Reader application or device must play the show preferentially in “watch” mode.

EXAMPLE:

In this example, a Video “show” rendition of a panorama sequence might show a pan-zoom sequence across a stitched version of the image sequence.

```
<mpv:AssetList>
  <mpv:StillPanoramaSequence mpv:id="ID000101">
  ...
  <mpv:Rendition mpv:renditionUsage="show">
    <mpv:VideoRef mpv:idRef="ID000102"/>
  </mpv:Rendition>
  ...
</mpv:StillPanoramaSequence>
  <mpv:Video mpv:id="ID000102">
  ...
  </mpv:Video>
</mpv:AssetList>
```

5.5 Practices for Asset Relationships

There are multiple types of relationships assets can have. MPV CE Playlists defines a basic set of relationships for interchange.

5.5.1 Asset Relationships

PLC100-14 Asset Relationships. Five categories of asset relationships are highlighted for interoperability: *Master asset*; *alt*, *thumbnail*, *screen*, and *subsampled Rendition*

assets; derivedFrom Related assets. Both Writers and Readers SHOULD support these relationships.

[MPVCore] defines a large number of renditions and several relationship types. For purposes of Playlist interoperability, only a subset of these are specified to be required for interoperability. These are:

- Master asset: The Master asset is the one referenced by the mpvp:Album playlist or determined to be a root asset. When the asset is a simple asset, the file containing the content of the primary or master asset is referred to by the mpv:LastURL element(s) of that asset. When choosing among multiple candidate files, the Primary asset SHOULD be the one that the user most naturally will think of as THE asset. For example, for images, the asset may be the highest resolution image
- The primary mpv:Still asset is one that element is one that is The Readers SHOULD first try to use the primary asset referenced by the mpv:LastURL value. However, when the asset cannot be found or the asset file format cannot be understood, the reader SHOULD use one of the alt Renditions, if present.

5.5.2 Master asset

PLC100-15 Master asset. Every MPV asset MUST have a master asset, which is a root asset in the asset list or in the mpvp:Album playlist and is the asset that conceptually is the center of the user's interaction. In other words, the Master asset corresponds most naturally with the user's conceptual model.

Typically, the Master asset will be the highest resolution asset or perhaps the asset with the highest level of file format compatibility.

5.5.3 Screen rendition

PLW100-10 Screen rendition. Every visual asset SHOULD have a screen rendition and non-visual assets MAY have a screen rendition. Screen renditions MUST be visual assets, e.g. mpv:Still, mpv:StillMultishotSequence, mpv:StillPanoramaSequence, mpv:Video, and the resolution MUST be at least corresponding to typical screen resolution among current products, such as VGA or 1MPixel 2D resolution.

Screen renditions are used for multiple purposes. They allow a reader to enhance performance and reduce memory usage by utilizing a lower-resolution still image than the master asset. For non-visual assets, screen renditions provide visual displays to accompany or stand-in for the non-visual assets. For a mpv:ManifestLink asset, the screen rendition shows a proxy image or video asset for the referenced manifest.

PLR100-3 Screen rendition. A reader MUST display screen renditions of assets when the reader device or application can do visual display at screen resolutions. When they are present, screen renditions MUST be displayed for the following asset types: mpv:Still, mpv:Video, mpv:Audio, mpv:ManifestLink.

5.5.4 Thumbnail rendition

PLW100-11 Thumbnail rendition. Every visual asset MUST have a thumbnail rendition and non-visual assets MAY have a thumbnail rendition. Thumbnail renditions MUST be simple visual assets, e.g. mpv:Still or mpv:Video, and the resolution MUST be less than VGA resolution.

Thumbnail renditions are used to allow a reader to enhance performance and reduce memory usage by utilizing a lower-resolution still image than the master asset. For non-visual assets, thumbnail renditions provide visual displays to accompany or stand-in for the non-visual assets. For a mpv:ManifestLink asset, the thumbnail rendition shows a proxy image or video asset for the referenced manifest.

PLR100-4 Thumbnail rendition. *A reader MUST display thumbnail renditions of assets when the reader device or application can do visual display at thumbnail resolutions. When they are present, thumbnail renditions MUST be displayed for the following asset types: mpv:Still, mpv:Video, mpv:ManifestLink.*

5.5.5 Show rendition: TBD

PLW100-12 Show rendition. *Every composite asset MAY have a Show rendition and non-visual assets MAY have a thumbnail rendition. Thumbnail renditions MUST be simple visual assets, e.g. mpv:Still or mpv:Video, and the resolution MUST be less than VGA resolution.*

Thumbnail renditions are used to allow a reader to enhance performance and reduce memory usage by utilizing a lower-resolution still image than the master asset. For non-visual assets, thumbnail renditions provide visual displays to accompany or stand-in for the non-visual assets. For a mpv:ManifestLink asset, the thumbnail rendition shows a proxy image or video asset for the referenced manifest.

PLR100-5 Show rendition. *A reader SHOULD display the Show rendition of assets when the reader device or application can do visual display at thumbnail resolutions. Thumbnail renditions SHOULD be displayed for the following asset types: mpv:Still, mpv:Video, mpv:ManifestLink.*

5.5.6 Example:

```
<mpv:Still mpv:id="ID000001">
...
  <mpv:Rendition mpv:renditionUsage="screen">
    <mpv:StillRef mpv:idRef="ID000001S"/>
  </mpv:Rendition>
  <mpv:Rendition mpv:renditionUsage="thumbnail">
    <mpv:StillRef mpv:idRef="ID000001T"/>
  </mpv:Rendition>
</mpv:Still>
<mpv:Still mpv:id="ID000001S">
...
  <mpv:LastURL mpv:filesystem="ISO9660-1">SCREEN/IMG0003.JPG</mpv:LastURL>
...
</mpv:Still>
<mpv:Still mpv:id="ID000001T">
...
  <mpv:LastURL mpv:filesystem="ISO9660-1">THUMB/IMG0003.JPG</mpv:LastURL>
...
</mpv:Still>
```

5.5.7 Alt rendition

PLR100-6 mpv:Still Renditions: alt. *Readers SHOULD first try to use the primary asset referenced by the mpv:LastURL value. However, when the asset cannot be found or the asset file format cannot be understood, the reader SHOULD use one of the alt Renditions, if present.*

This practice gives the reader additional choices when seeking to display an asset file. A typical usage will be to

EXAMPLE:

```
<mpv:Still mpv:id="ID000001">
...
  <mpv:Rendition mpv:renditionUsage="screen">
    <mpv:StillRef mpv:idRef="ID000001S"/>
  </mpv:Rendition>
  <mpv:Rendition mpv:renditionUsage="thumbnail">
    <mpv:StillRef mpv:idRef="ID000001T"/>
  </mpv:Rendition>
</mpv:Still>
<mpv:Still mpv:id="ID000001S">
...
  <mpv:LastURL mpv:filesystem="ISO9660-1">SCREEN/IMG0003.JPG</mpv:LastURL>
...
</mpv:Still>
<mpv:Still mpv:id="ID000001T">
...
  <mpv:LastURL mpv:filesystem="ISO9660-1">THUMB/IMG0003.JPG</mpv:LastURL>
...
</mpv:Still>
```

5.5.8 Subsampled rendition

5.5.9 DerivedFrom Related Asset

5.5.10 CopyOf Related Asset

5.6 *Practices for Asset mpv:LastURLs*

5.6.1 mpv:filesystem attribute

PLW100-13 mpv:LastURL mpv:filesystem attribute. Each asset that has at least one mpv:LastURL element must provide at least one mpv:LastURL element for each filesystem of the storage media on which the MPV playlist is resident and with which the asset can be accessed.

This practice is to guarantee that a reader can find a mpv:LastURL element with the mpv:filesystem attribute that corresponds to the active filesystem.

The mpv:filesystem attribute is used on mpv:LastURL elements to identify the filesystem with which the LastURL value is associated. The LastURL values can be different for different filesystems because filesystems have limits

such as number of characters in each pathname segment. For CDs in particular, it is commonplace to have multiple filesystems on a CD, such as the ISO-9660-1 filesystem and the Joliet filesystem.

5.6.2 UTF-8 and URL-encoding

PLW100-14 mpv:LastURL UTF-8 and URL-encoded values. The value of a mpv:LastURL element MUST be UTF-8-encoded and URL-encoded.

This practice is to guarantee that a lastURL value is properly decoded.

A LastURL value is always UTF-8 and URL-encoded. The algorithm for generating a LastURL value is as follows:

1. UTF-8 encode the pathname.
 - a. Converts multibyte or Unicode strings into UTF-8 strings
 - b. UTF-8 encoding is required by MPV Manifests
 - c. Note that filesystem-specific practices are retained, e.g. MS-DOS and MS-Windows directory separator backslashes “\” and Macintosh directory separators colons “:”. This is appropriate if the target filesystem of the lastURL value is the same as the filesystem from which the pathname was derived. However, when converting between filesystems, e.g. from a harddisk filesystem like NTFS to a CD filesystem like ISO9660-1, additional processing may be needed to convert directory separators and pathname components to be compliant with the target filesystem.
2. URL-encode the pathname.
 - a. Converts all URL-protected characters into equivalents, such as space “ ” into %20.

The resulting value should be transportable and processable in any environment, since it uses only a 7-bit character set.

To convert the LastURL value back to a value that can be used by the filesystem, reverse the algorithm.

1. URL-decode the pathname.
2. UTF-8 decode the pathname.

The resulting value should be correct for the given filesystem indicated by the mpv:filesystem attribute.

5.6.3 Fragment Identifier Processing

PLR100-7 mpv:LastURL Fragment Identifier Processing. Fragment identifiers in mpv:LastURL values MUST be allowed and handled gracefully, even if the specific fragment identifier understood.

This practice guarantees that techniques such as identifying thumbnails resident in JPEG Exif 2.1+ files using fragment identifiers will be supported by MPV CE Playlist players. If a specific fragment identifier is not understood, the player may choose to process the asset as an unknown type.

5.7 Practices for mpv:Still

5.7.1 Image Size metadata

PLW100-15 mpv:Still Image size metadata. Every still image MUST have the image size metadata.

This practice guarantees that the reader can determine the still image size without opening the media file. This is important on low-performance systems and also when MPV is used over an internet connection with a web browser. In particular, the web browser environment does not allow direct access to the image file itself to recover size information, but size information may be critical to the correct operation of the webpage or webserver functionality,

such as being able to recommend that low-resolution still images not be printed larger than a certain size. The metadata to use for image size is the JPEG2000-based schema as defined by example in [MPVCore].

EXAMPLE:

```
<mpv:Still>
  <mpv:ContentID>urn:osta-
  org:mpv:dsig:md5:all:EF886AEFA3B340da971BAF09B17DBC12</mpv:ContentID>
  <mpv:LastURL mpv:filesystem="ISO9660-1">IMG0003.JPG</mpv:LastURL>
  <mpv:Metadata>
    <BASIC_IMAGE_PARAM xmlns="http://www.jpeg.org/jpx">
      <BASIC_IMAGE_INFO>
        <IMAGE_SIZE>
          <WIDTH>1600</WIDTH>
          <HEIGHT>1200</HEIGHT>
        </IMAGE_SIZE>
      </BASIC_IMAGE_INFO>
    </BASIC_IMAGE_PARAM>
  </mpv:Metadata>
</mpv:Still>
```

5.7.2 mpv:Still Renditions: thumbnail, screen

PLR100-8 mpv:Still Renditions: thumbnail, screen. When present, readers SHOULD use thumbnail and screen renditions of mpv:Still when displaying thumbnail and screen resolution images because they minimize memory usage and enhance performance.

This practice allows the reader to more rapidly display image content and hold more images in memory.

EXAMPLE:

```
<mpv:Still mpv:id="ID000001">
  ...
  <mpv:Rendition mpv:renditionUsage="screen">
    <mpv:StillRef mpv:idRef="ID000001S"/>
  </mpv:Rendition>
  <mpv:Rendition mpv:renditionUsage="thumbnail">
    <mpv:StillRef mpv:idRef="ID000001T"/>
  </mpv:Rendition>
</mpv:Still>
<mpv:Still mpv:id="ID000001S">
  ...
  <mpv:LastURL mpv:filesystem="ISO9660-1">SCREEN/IMG0003.JPG</mpv:LastURL>
  ...
</mpv:Still>
<mpv:Still mpv:id="ID000001T">
  ...
  <mpv:LastURL mpv:filesystem="ISO9660-1">THUMB/IMG0003.JPG</mpv:LastURL>
  ...
</mpv:Still>
```

5.7.3 mpv:Still Renditions: alt

PLR100-9 *mpv:Still Renditions: alt. Readers SHOULD first try to use the primary asset referenced by the mpv:LastURL value. However, when the asset cannot be found or the asset file format cannot be understood, the reader SHOULD use one of the alt Renditions, if present.*

This practice gives the reader additional choices when seeking to display an asset file. A typical usage will be to

EXAMPLE:

```
<mpv:Still mpv:id="ID000001">
...
  <mpv:Rendition mpv:renditionUsage="screen">
    <mpv:StillRef mpv:idRef="ID000001S"/>
  </mpv:Rendition>
  <mpv:Rendition mpv:renditionUsage="thumbnail">
    <mpv:StillRef mpv:idRef="ID000001T"/>
  </mpv:Rendition>
</mpv:Still>
<mpv:Still mpv:id="ID000001S">
...
  <mpv:LastURL mpv:filesystem="ISO9660-1">SCREEN/IMG0003.JPG</mpv:LastURL>
...
</mpv:Still>
<mpv:Still mpv:id="ID000001T">
...
  <mpv:LastURL mpv:filesystem="ISO9660-1">THUMB/IMG0003.JPG</mpv:LastURL>
...
</mpv:Still>
```

5.7.4 mpv:Still Derivations

PLR100-10 *mpv:Still Renditions. Readers must support two types of mpv:renditionUsage for mpv:Still: thumbnail, screen*

This practice guarantees that the reader can determine the still image size without opening the media file. This is important on low-performance systems and also when MPV is used over an internet connection with a web browser. In particular, the web browser environment does not allow direct access to the image file itself to recover size information, but size information may be critical to the correct operation of the webpage or webserver functionality, such as being able to recommend that low-resolution still images not be printed larger than a certain size. The metadata to use for image size is the JPEG2000-based schema as defined by example in [MPVCore].

5.8 Practices for MPV Basic Spec Usage

TODO

5.9 Practices for MPV Presentation Spec Usage

5.9.1 Attribute `mpv:manifestLinkIdRef` NOT USED

PLC100-16 Attribute `mpv:manifestLinkIdRef`. The attribute `mpv:manifestLinkIdRef` MUST NOT be used on any element.

All the assets referenced by the `<mpvp:Album>` must be present in the `<mpv:AssetList>` within the same MPV manifest. This is enforced by disallowing use of the `mpv:manifestLinkIdRef` attribute on any asset reference element.

5.9.2 Only one `<mpvp:Album>` element

PLC100-17 Only one `<mpvp:Album>`. Zero or one `<mpvp:Album>` elements MUST be present, but not more..

The [MPV Pres] specification allows for more than one `<mpvp:Album>` element in a MPV manifest. The MPV CE Playlists specification requires that there is at most one `<mpvp:Album>` element. This restriction means that the words “MPV Album” and “MPV Manifest” mean the same thing – a single set of assets, optionally containing presentation information.

5.9.3 MUST NOT: Use `<mpvp:AlbumRef>` to link to another Album

PLC100-18 `<mpvp:AlbumRef>` NOT USED. The `<mpvp:AlbumRef>` MUST NOT be used.

The use of the `<mpvp:AlbumRef>` asset reference is not allowed. Instead, use `<mpv:ManifestLinkRef>`. A `<mpv:ManifestLink>` asset is used to link one manifest to another.

5.10 Practices for MPV Music Profile Spec Usage

TODO

5.11 Practices for MPV Extensions

MPV allows applications and devices to create custom extensions to MPV and store them as an integrated part of a MPVmanifest. MPV CE Playlists supports this practice. The following practices dictate how extensions are handled.

TODO

Chapter 6: MPV CE Playlist Playback Practices

This section will define a standard set of player actions and the expected practices.

Track Playback operations:

Play	Play the current playlist media item
Stop	Stop playback of the current playlist media item
Seek	Move the playback cursor forward or backward within the current playlist media item.

Track Navigation operations:

Next	Play the next playlist media item
Previous	Play the previous playlist media item

Playlist Navigation operations:

Show Playlist	Show the contents of the playlist media item
LinkTo Playlist	Make the playlist at the selected link to the current playlist
Previous Playlist	Make the previous playlist the current playlist

Player operations:

Activate Player	Activate the MPV player
Detect Playlist	Detect presence of a MPV playlist

Lots more TODO here.

Appendix I: References

[DATETIME]

"Date and Time Formats", M. Wolf, C. Wicksteed. W3C Note 27 August 1998,
Available at: <http://www.w3.org/TR/NOTE-datetime>

[DC]

"Dublin Core Metadata Initiative", a Simple Content Description Model for Electronic Resources.
Available at <http://purl.org/DC/>

[DC-NMF]

"Dublin Core Normalized Metadata Format Profile Specification 1.0"; OSTA, 2002.
Available at <http://www.osta.org/mpv/>

[DCF-1999]

"Design rule for Camera File system, Version 1.0", JEIDA standard, English Version 1999.1.7, Japanese Electronic Industry Development Association (JEIDA).

[DIG35-2001]

"DIG35 Specification – Metadata for Digital Images, Version 1.1", June 18, 2001, International Imaging Industry Association (I3A) [recently formed by combining the Digital Imaging Group and PIMA].
<http://www.i3a.org>

[ISO8601]

"Data elements and interchange formats - Information interchange - Representation of dates and times", International Organization for Standardization, 1998.

[ISO10646]

""Information Technology -- Universal Multiple-Octet Coded Character Set (UCS) -- Part 1: Architecture and Basic Multilingual Plane", ISO/IEC 10646-1:1993. This reference refers to a set of codepoints that may evolve as new characters are assigned to them. This reference therefore includes future amendments as long as they do not change character assignments up to and including the first five amendments to ISO/IEC 10646-1:1993. Also, this reference assumes that the character sets defined by ISO 10646 and Unicode remain character-by-character equivalent. This reference also includes future publications of other parts of 10646 (i.e., other than Part 1) that define characters in planes 1-16. "

[JFIF]

"JPEG File Interchange Format, Version 1.02"; Eric Hamilton, September 1992.
Available at <http://www.w3.org/Graphics/JPEG/jfif.txt>

[MANIFEST]

"XML Manifest Specification 1.0"; OSTA, 2002.,
Available at <http://www.osta.org/mpv/>

[MD5]

"The MD5 Message-Digest Algorithm", RFC 1321, April 1992.
Available at <http://www.ietf.org/rfc/rfc1321.txt>. Further information and source code available at <http://userpages.umbc.edu/~mabzug1/cs/md5/md5.html>

[MIME-2]

"RFC 2046: Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types"; N. Freed, N. Borenstein, November 1996.
Available at <ftp://ftp.isi.edu/in-notes/rfc2046.txt>

[MIMETYPES-REG]

IANA official registry of MIME media types
Available at <http://www.isi.edu/in-notes/iana/assignments/media-types/media-types>

[MPVBasic]

"MPV – Basic Profile Specification", OSTA, 2002,
Available at <http://www.osta.org/mpv/>

[MPVCore]

"MPV Core Specification 1.0"; OSTA, 2002.,
Available at <http://www.osta.org/mpv/>

[MPVDoc]

"MPV Document Profile Specification 1.0"; OSTA, 2003.,
Available at <http://www.osta.org/>

[MPVPres]

"MPV Presentation Profile Specification 1.0"; OSTA, 2002.,
Available at <http://www.osta.org/mpv/>

[NMF]

"Normalized Metadata Format Specification 1.0"; OSTA, 2002.,
Available at <http://www.osta.org/mpv/>

[PNG-MIME]

"Registration of new Media Type image/png"; Glenn Randers-Pehrson, Thomas Boutell, 27 July 1996.
Available at <ftp://ftp.isi.edu/in-notes/iana/assignments/media-types/image/png>

[PNG-REC]

"PNG (Portable Network Graphics) Specification Version 1.0"; Thomas Boutell (Ed.).
Available at <http://www.w3.org/TR/REC-png>

[QT]

"QuickTime Movie File Format Specification", May 1996.
Available at <http://developer.apple.com/techpubs/quicktime/qtdevdocs/REF/refFileFormat96.htm>

[QT-MIME]

"Registration of new MIME content-type/subtype"; Paul Lindner, 1993.
Available at <http://www.isi.edu/in-notes/iana/assignments/media-types/video/quicktime>

[RDFsyntax]

"Resource Description Framework (RDF) Model and Syntax Specification", Ora Lassila and Ralph R. Swick. W3C Recommendation 22 February 1999, Available at <http://www.w3.org/TR/REC-rdf-syntax/>

[RDFschema]

"Resource Description Framework (RDF) Schema Specification", Dan Brickley and R.V. Guha. W3C Proposed Recommendation 03 March 1999, Available at <http://www.w3.org/TR/PR-rdf-schema/>

[RFC1766]

"Tags for the Identification of Languages", H. Alvestrand, March 1995. Available at <ftp://ftp.isi.edu/in-notes/rfc1766.txt>

[RFC2119]

"Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, IETF RFC 2119 <http://www.ietf.org/rfc/rfc2119.txt>

[URI]

"Uniform Resource Identifiers (URI): Generic Syntax", T. Berners-Lee, R. Fielding, L. Masinter, August 1998. Note that RFC 2396 updates [RFC1738] and [RFC1808].

[UCS-2]

16-bit encoding of ISO 10646, commonly known as the Unicode character set.

[UTF-8]

Yergeau, F., "UTF-8, a transformation format of ISO 10646", RFC 2279, January 1998.

[W3C-NSURI]

"URIs for W3C namespaces". Policy and administrative issue for W3C, Oct. 1999. Available at <http://www.w3.org/1999/10/nsuri>

[XML10]

"Extensible Markup Language (XML) 1.0" T. Bray, J. Paoli and C.M. Sperberg-McQueen. W3C Recommendation 10 February 1998, Available at <http://www.w3.org/TR/REC-xml>

[XML-NS]

"Namespaces in XML", Tim Bray, Dave Hollander, Andrew Layman. W3C Recommendation 14 January 1999, Available at <http://www.w3.org/TR/REC-xml-names>

[XSCHEMA]

"XML Schema, XML Schema Part 1: Structures". W3C Working Draft, work in progress. Available at <http://www.w3.org/TR/xmlschema-1/>

[XSL]

"Extensible Stylesheet Language (XSL) Specification", Stephen Deach. W3C Working Draft, work in progress. Available at <http://www.w3.org/TR/xsl/>