MultiPhoto/Video

Manifest, Metadata and Practices for Digital Photo-Video Collections

DRAFT LOGO

Presentation Profile Specification

Revision 0.39
Working Draft

May 22, 2002

IMPORTANT NOTICE

This document is a working draft for review by OSTA and I3A 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 and I3A 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-Spec-Presentation-0.39WD.PDF

POINTS OF CONTACT

OSTA David Bunzel OSTA President Tel: +1 (408) 253-3695 Email: dbunzel@osta.org http://www.osta.org *I3A* Lisa Walker I3A Co-Executive Director and Chief Marketing Officer Tel: +1 949-481-7645 Email: lisaw@i3a.org http://www.i3a.org	*MultiPhoto/Video Website* http://www.osta.org/mpv/index.htm *Technical Content* Pieter van Zee Editor, MultiPhoto/Video Specification Tel: +1 541-715-8658 Email: pieter_van_zee@hp.com Felix Nemirovsky Chairman, MultiRead Subcommittee Tel: +1 415 643 0944 Email: felixn@oaktech.com

ABSTRACT

The MultiPhoto/Video specification defines a manifest and metadata format and practices for processing and playback of collections of digital photo, video, and related audio and file content stored on an optical disc and other storage media such as memory cards and computer harddrives or exchanged via internet protocols.

COPYRIGHT NOTICE

LICENSING
IMPORTANT NOTICES

This document was developed by the Optical Storage Technology Association (OSTA) and International Imaging Industry Association (I3A). This document may be revised by OSTA. It is intended solely as a guide for companies interested in developing products which can be compatible with other products developed using this and closely-related documents. OSTA makes no representation or warranty regarding this document, and any company using this document 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 of this document. This document defines only one approach to compatibility, and other approaches may be available in the industry.

This document is an authorized and approved publication of OSTA and I3A. The underlying information and materials contained herein are the exclusive property of OSTA and of the I3A as defined by a separate license but may be referred to and utilized by the general public for any legitimate purpose, particularly in the design and development of optical recording and reading 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. Only OSTA has the right and authority to revise or change the material contained in this document, and any revisions by any party other than OSTA are totally unauthorized and specifically prohibited, except as specifically allowed by a license from the OSTA.

Compliance with this document may require use of one or more features covered by proprietary rights (such as features which are the subject of a patent, patent application, copyright, mask work right or trade secret right). By publication of this document, no position is taken by OSTA or I3A with respect to the validity or infringement of any patent or other proprietary right, whether owned by a Member or Associate of OSTA or otherwise. OSTA and I3A hereby expressly disclaim any liability for infringement of intellectual property rights of others by virtue of the use of this document. OSTA and I3A have not and do not investigate any notices or allegations of infringement prompted by publication of any OSTA or I3A document, nor does OSTA or I3A undertake a duty to advise users or potential users of OSTA and I3A documents of such notices or allegations. OSTA and I3A 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 and I3A expressly disclaim any intent to promote infringement of any intellectual property right by virtue of the evolution, adoption, or publication of this document.

The information in this document is believed to be accurate as of the date of publication.
THIS DOCUMENT IS PROVIDED "AS IS." THE OPTICAL STORAGE TECHNOLOGY ASSOCIATION AND INTERNATIONAL IMAGING INDUSTRY ASSOCIATION MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO: WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THIS DOCUMENT ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.
THE OPTICAL STORAGE TECHNOLOGY ASSOCIATION AND INTERNATION IMAGING INDUSTRY ASSOCIATION WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE DOCUMENT OR THE PERFORMANCE OR IMPLEMENTATION OF THE CONTENTS THEREOF.

MultiPhoto/Video is a trademark of Optical Storage Technology Assocation, Inc. All other trademarks are the property of their respective owners. The names and/or trademarks of OSTA and I3A members may NOT be used in advertising or publicity pertaining to this document or its contents without specific, written prior permission.

Contents

Chapter 1: Introduction

1.1 Executive Summary

MultiPhoto/Video (MPV) is an open specification that makes easier the processing and playback of collections of photo-video content, including stills, stills with audio, still sequences, video clips, and audio clips. By analogy, MPV is added to the original data to enable slideshow and browsing tasks of photo-video content just as DPOF is added to the original data to enable printing of photo content.

Applications and devices and users that use MultiPhoto/Video benefit even when they only interact with still images in basic ways; when content like video clips and still sequences are added, as can be captured by a majority of the digital cameras introduced recently, the benefits expand.

MultiPhoto/Video uses a simple text-based format that is easily understood and also easy to produce and consume programmatically in firmware or computer software. MultiPhoto/Video does *not* tackle a large number of problems at once – instead, it focuses on a few key problems that it solves with simple but robust approaches. Where possible and practical, it makes use of established specifications and standards.

The development and promotion of MultiPhoto/Video is sponsored jointly by two industry-leading trade associations, the Optical Storage Technology Assocation (OSTA) and the International Imaging Industry Association (I3A). The specification development and promotion process is open to all members; all organizations and individuals are welcomed as members. These associations include over 100 member companies from all over the world that produce products that collectively represent a majority marketshare in mainstream consumer digital imaging and recordable optical storage categories.

MultiPhoto/Video is not only a specification. It also includes a compliance test suite and processes, compliance testing materials, and a logo program for compliant products. In addition, some sample open-source code implementations of key steps in processing MPV content are available. These materials and procedures are made available and administered by OSTA at a modest cost. OSTA and I3A charge no royalty for use of the specification or logo.

The specification is being developed in phases and results in "profiles". Each profile in MultiPhoto/Video defines only those formats and practices that are necessary for the key tasks targeted by the profile. A number of candidate profiles for development have been identified, 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
- **Internet Profile**: key task: interacting with and sending collections of photo-video content over the web and email
- **Capture Profile**: key task: writing new content to storage media and updating the collection info
- **Disc Archive Profile**: key task: interoperability of photo archives on recordable optical discs
- **Editing Profile**: key task: modifying existing collections of photo-video content.

- **Printing Profile:** key task: printing collections of photo-video content
- **Container Profile**: key task: storing photo-video content collections in containers

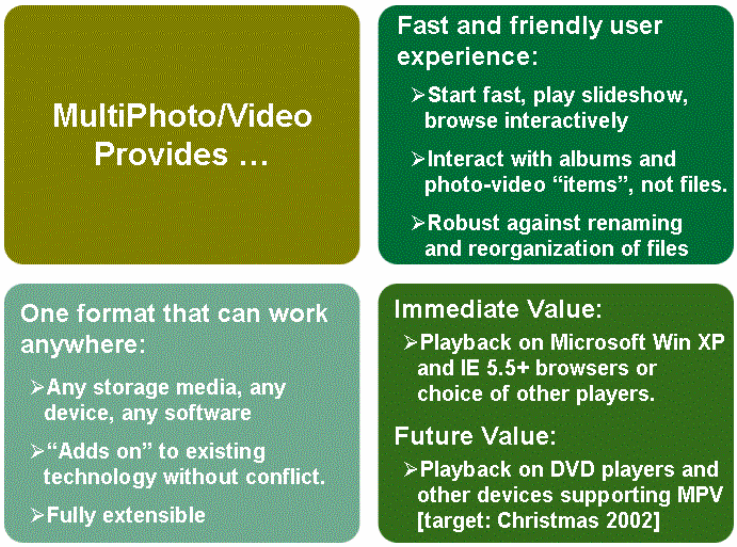
Underlying all profiles is the "Core Module", which defines the overall framework of all MPV profiles. The Basic and Presentation Profiles, for example, build on the Core Module and, when implemented in consumer electronics devices like DVD players or in application software, can provide compelling playback of photo-video slideshows and interactive browsing of photo-video content.

MultiPhoto/Video technology has three central components: Collections, Metadata, and Identification. Each of these make reference in various ways to data files containing the photo-video content. This information is augmented with Presentation information that may be used by player applications and devices to provide an attractive user experience.

1.2 Overview

MultiPhoto/Video (MPV) is an open specification to enhance interoperability, ease-of-use, and abilities to play and manipulate collections of photo/video content, including still images, still with audio, still sequences, video clips, audio-only clips, and related files. MPV is made available at low cost and without royalty from the Optical Storage Technology Assocation (OSTA) and the International Imaging Industry Association (I3A). OSTA is an industry association promoting the use and interoperability of recordable CD and DVD discs in computer and consumer electronics devices. I3A is an industry association promoting digital and film imaging technologies.

MPV enables PC software and consumer electronics devices like DVD players to playback and manipulate collections of digital photo/video content including still images, still with audio, still sequences, video clips, audio-only clips, and related files. The emphasis is on personal content originating from many sources including digital cameras, film, scanners and video digitizer and stored on a range of media including memory cards, recordable or stamped CDs and DVDs, and even computer harddisks or internet services.



Development of the specification will be in multiple stages. A basic profile for use by DVD players and media player software to provide slideshows and interactive browsing of digital photo/video content will be completed first – that is this document. Another basic profile for photo-video capture products like digital cameras, scanners, and imaging software will be developed subsequently. Both profiles will be fairly simple and easy to support.

The MPV specification will further promote adoption of current and new categories of digital imaging products by enhancing ease-of-use and interoperability of photo/video content collections and applications. The format enables an end-user experience that starts fast, is highly interactive, provides for playing and editing collections of photo/video content, never reveals the underlying storage file system, and can be implemented in firmware of consumer electronics devices like DVD players as well as by PC software. MPV can be produced automatically or interactively by digital cameras, scanners, imaging software, internet services and other devices.

MPV provides specific manifest and metadata formats and implementation practices that support existing industry specifications such as the World Wide Web Consortium's SMIL, I3A's DIG35, and Adobe's eXtensible Metadata

Platform XMP. MPV is compatible with and supports the DCF and Exif specifications from the JEITA and JCIA that are widely used in digital cameras. New metadata elements will be developed as necessary. The work is oriented to deliver tangible and useful results in the near-term.

Support for MPV can be "added on" to existing applications and conventions because it is non-invasive and can co-exist with existing file system structures and formats. The format is designed for longevity and extensibility through the use of industry-standard XML. The manifest format will support write-only media, high-performance update, and use in low-memory, low-performance devices.

Key technical advances provided by the MPV specification specifically enable or enhance interoperability and end-user experience. Collections of photo-video content can be specified with optional presentation information. Practices for how to represent, compute, insert, and compare identifiers of digital assets enable collections to be more robust when assets are renamed or moved. Metadata for compound assets like still image sequences and primary and dependent assets (e.g. thumbnails, low-res renditions) allow manipulation of higher level constructs than the individual primary assets.

The MPV format does not contain the content itself -- MPV is an aggregation of information about the content, including references to the content. It provides essentially a Table of Contents and metadata repository; a typical implementation is a stand-alone file such has "ALBUM.MPV" and zero or more dependent files.

MPV is well suited as an intermediate format for exchange of photo-video content collections across applications, devices, and services. Some applications may also choose to use it as the primary format for storing their own data. MPV is structured such that it may be used with reasonable efficiency as a lightweight textual database to maintain metadata and related information for hundreds to the low thousands of photo-video content files.

1.3 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 MultiPhoto/Video specification document for a precise and legal description.

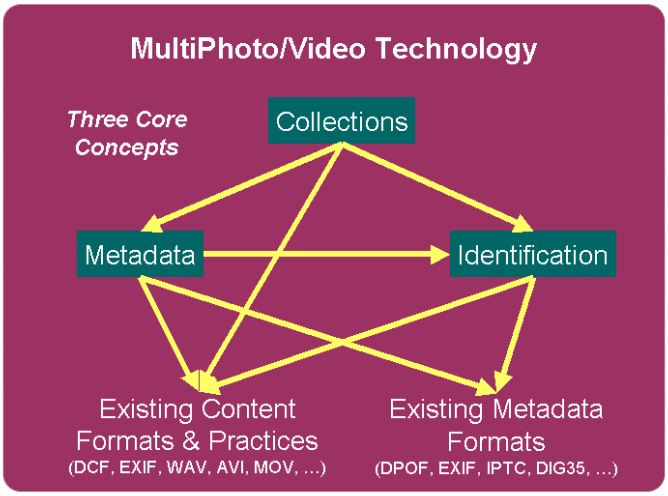
The MultiPhoto/Video specification is developed using an open process. The resulting specification is available at no or modest cost from OSTA and I3A. No royalty is charged by OSTA or I3A for use of the specification. The overall desire is to develop a specification that is not subject to separate licensing 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 and I3A does not warrant that the specification is not or will not be subject to such claims by other parties.

MultiPhoto/Video is not only a specification. It also includes a compliance test suite and processes, compliance testing materials, and a logo program for compliant products. In addition, some sample open-source code implementations of key steps in processing MPV content are available. These materials and procedures are made available and administered by OSTA at a modest cost. OSTA and I3A charge no royalty for use of the specification or logo.

Chapter 2: Key Concepts

MultiPhoto/Video has some key concepts and approaches.

- The Basic Profile has three core concepts centered on Collections, Metadata, and Identification.
- The Presentation Profile adds in the Album view of the collection and other Presentation information.
- Profiles and Modules structure the specification, while the Schema formally defines it and Practices guide its use.



2.1 Collections

Collections are assembled using a few core concepts.

MANIFEST

The MPV manifest groups all the MPV components into a single XML document. A MPV manifest contains a least one asset list or manifest links. It may contain zero or more albums and mark lists, which provide views onto those assets. In typical usage, a MPV manifest is stored in a stand-alone file.

ASSET LIST

An asset list is an unordered set of assets that each have a unique local identifier in the MPV collection. It is the only place photo-video assets may be defined as part of the collection – everything else in MPV is metadata and references to assets. A MPV collection contains at least one asset list or link to a manifest in another file. By analogy, an asset list may be considered a table of assets in a database and the id is the foreign key. Another analogy would be to the entries in a Unix file system inode.

MARK LIST

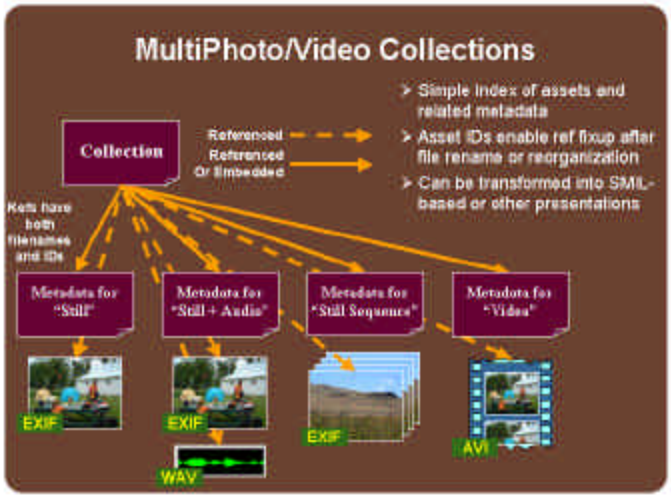
A mark list is an ordered set of asset references and associated metadata and mark type. A MPV collection may contain zero or more mark lists. The optional mark list with the special "primary" mark type identifies which assets in the asset list are considered to be top-level assets in a collection and gives them an order. Other predefined mark types are "selected" and "hidden"; the mark type is fully extensible.

SIMPLE MEDIA ASSETS

An asset list may contain the following types of media assets. MPV does not constrain which formats of these media assets may be in a collection. Simple media assets correspond to physical storage entities, i.e. files.

- Still
- Video
- Audio
- Text
- Print
- Document
- ManifestLink

Any media asset may contain renditions and related documents.



COMPOSITE MEDIA ASSETS

In addition to the simple media assets, MPV also defines composite assets, which are semantically meaningful groups of media assets. These correspond to typical capture modes of digital cameras.

- StillWithAudio
- StillMultishotSequence
- StillPanoramaSequence
- Par
- Seq

Composite media assets may be primary assets, renditions, or related documents. The Seq and Par assets allow for arbitrary expression of other media assets but lack the direct association with the user's capture mode.

RENDITIONS

Any simple or composite media asset and even an album may have one or more renditions. Typically, original asset is the master rendition and is usually defined implicitly. Renditions other than the master rendition are derived versions of the original media asset. The relationship between the original rendition and the derived renditions is captured in metadata. The derived version may be direct, as in a screen resolution image of a hi-res image, or indirect, as in a video stream or print rendition of a collection.

RELATED DOCUMENTS

All simple and composite media assets and an album may have one or more related documents. Such documents may have any relation to the media asset, including other assets used in constructing the asset or additional metadata related to the asset.

2.2 Metadata

MPV IS METADATA, NOT DATA

MPV provides metadata to describe photo/video asset collections. It does not contain the actual asset data files themselves. The set of MPV metadata defines collections, identifiers, simple and composite assets, and a basic set

of presentation information. MPV also provides the ability to embed completely arbitrary XML-formatted metadata from any source, providing an easy and open extension mechanism.

MPV provides full interoperability with Adobe System's open metadata specification called XMP (Extensible Metadata Platform), a rich family of metadata schema and practices for individual assets of many types that is being adopted by many commercial vendors. This is the preferred mechanism to specify many kinds of common metadata in MPV, such as for Dublin Core, Graphics, Image, Dynamic Media, Video, Audio, Text, PagedText, Rights management, and Media management. Using the VXMP framework utilized by MPV, custom metadata schema can be designed that is fully interoperable with both MPV and XMP and also fully validatable using commonly available XML-Schema-based tools.

OTHER METADATA

Generally speaking, MPV recommends that metadata about basic media assets be embedded in the asset. Recommended practices are provided for using existing metadata formats in typical media file formats, such as Exif, JFIF, TIFF, WAV, MP3, MPG, AVI, and MOV. Metadata for composite media assets often cannot reside only in the basic media assets because it spans multiple asset files. This information is often stored in various established metadata formats such as I3A's DIG35 and Adobe's XMP. This type of metadata may be embedded within an MPV document, even when it is not part of the MPV schema.

XML PACKETS

MPV uses XML packets to provide for embedding and extracting MPV metadata in arbitrary files. The XML packet format is defined by Adobe's XMP specification.

NAMESPACES AND NAMING EQUIVALENCE

XML namespaces are a means to allow elements of the same name that exist in different schema to co-exist within the same document.

MPV requires that the MPV namespace prefixes on all elements and attributes be used in all XML encodings.

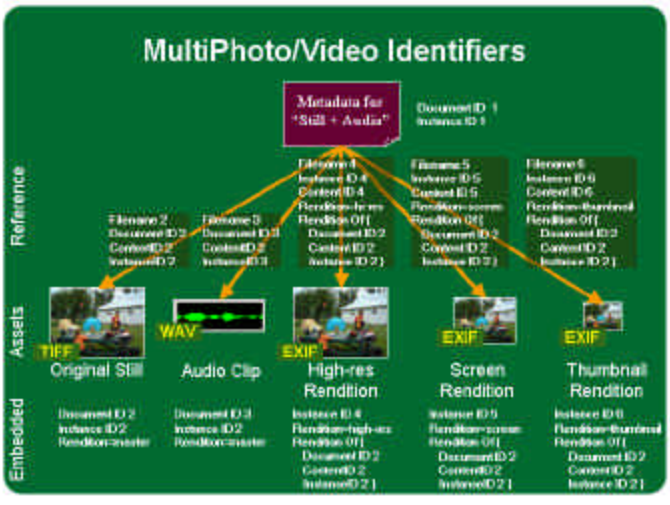
Some older existing XML-based applications and schema do not support namespaces. MPV can be encoded using a pseudo namespace by prefixing all elements and attributes with a defined namespace prefix separated with the underscore ("_"). Such an encoding is defined to enable the MPV specification to be used when namespaces are not supported; however, documents of this type are NOT well-formed MPV documents and need to be translated to use namespaces before they can be expected to interoperate with other MPV processors.

2.3 Identifiers

TYPES OF IDENTIFIERS

Identifiers are the means by which references are made between a collection and the assets it references. All basic and composite media assets in a collection are identified by two or more identifiers. There are four kinds of identifiers:

- id – a unique identifer local to the MPV collection in which it is used and used to reference elements in a MPV collection.
- lastURL – last known location
- instanceID – unique identifier for an asset



- documentID – the same for all renditions
- contentID – computed using the concent as input; statistically unique for each asset.

More than one of the lastURL, docmentID, and contentID identifiers may be used. For example, multiple lastURLs may be provided to allow for different filenames in different file systems, such as on a CD. Multiple contentIDs may be provided that utilize different computation algorithms with various tradeoffs of speed and robustness.

The lastURL can be a local filename or remote URL. Significantly, lastURL is not a robust reference; it is broken easily by the user renaming or rearranging the referenced assets. Equally, the lastURL can be broken easily when a collection and assets are transferred across devices, storage formats and file systems.

To be robust against broken lastURL names, MPV provides identifier mechanisms and practices that allow the lastURL values to be fixed up when broken by searching for files with identifiers that match those contained in the collection. The ability to fixup broken references is a key contribution that MPV makes to industry practices for representing collections.

COMPUTING IDENTIFIERS

Identifiers can be computed and inserted in media assets in a variety of ways.

- arbitrary identifiers – computed in some manner independent of the asset data and assigned to the asset. Arbitrary identifiers are typically quick to generate and compare but are fragile because if they are damaged or lost, they cannot be reconstructed.
- content-based identifiers – computed in some manner dependent on the asset data. Content-based identifiers are typically slower to generate and compare, but are more robust and also less invasive because they can be regenerated based on the content itself.

Arbitrary identifiers are computing using a variety of algorithms typically available in the operating system. MPV uses the UUID 128-bit identifier which is readily generated by most modern operating systems. Sample source code for computing an assigned identifer is provided and can be used for firmware implementations.

Many content-based identifier computation methods exist. MPV specifies the MD5 algorithm as the basic algorithm that should always be supported. MD5 computes a 128-bit hash of the byte values in an arbitrary set of content.

2.4 Presentation

The MPV Basic Profile defines how to represent collections. The MPV Presentation Profile defines how to present them.

ALBUM, ALBUMLINK

An album is a presentation-oriented view of the asset list and the most common representation of an MPV collection exposed to users . It is an ordered set of references to assets in asset lists. Albums can link to other albums. Multiple albums can be grouped together in one file or isolated in separate files. Album links use URIs, allowing reference to local or remote albums. Albums may have renditions, related documents and mark lists of their own.

FOREGROUND, BACKGROUND

Users interact with Album-level Foreground and Background assets; they and the Album's Related Documents are conceptually the primary assets in a collection. Typically, users interact most with foreground assets while background assets are secondary and fewer. Foreground and background assets may also contain additional content, including renditions and related documents. Additional content may enhance the performance, scope, presentation, and other characteristics of an album but do not fundamentally change it from a user's perspective.

USER TASKS

Primary user tasks for albums are to allow the user to play a slideshow of or interactively browse the primary assets in the album. The MPV Presentation Profile extends the spec with very basic presentation information to enhance the user's experience.

PRESENTATION CONTROL

The overall approach for representing presentation information derives from SMIL, a powerful XML format for representing presentations from the World Wide Web Consortium (W3C). MPV Presentation Profile is a very constrained derivative of SMIL that provides just a basic level of presentation control. A MPV document can be mechanically translated into any of the common SMIL profiles. This makes MPV a good intermediate representation and also suggests a MPV playback strategy on platforms that also have SMIL players.

Because MPV also allows arbitrary metadata to be embedded or referenced, it is possible to embed additional presentation information in SMIL or other presentation languages. These may be used by players aware of these formats and practices.

XML LEVERAGE

MPV is well-formed XML. This allows the MPV album document to be used with standard XML processing environments. For example, when opened in the Microsoft Internet Explorer 5.5 and above web browser, an MPV document with associated style sheet can present an attractive user interface for playback of MPV photo-video collections. Similarly, straight forward XSLT translation can convert an MPV document into a SMIL-based presentation for playback with an appropriate player.

2.5 Profiles and Modules, Schema and Practices

The MultiPhoto/Video specification is organized in the following ways.

Schema define the structure of MPV content, proviing a precise grammar and vocabulary of expression. MPV uses XML-Schema [XSCHEMA], a well-known schema definition language, to define this grammar and vocabulary in combination with prose descriptions to clarify usage and behaviour. A wide variety of commercial and open source tools support the use of XML Schema, including for schema design and schema and content validation.

In MPV, all schema are available in machine-readable form in addition to inclusion on a fragmentary basis within the specification document. The machine-readable schema in the normative definition; in the case of discrepancy, it supercedes the fragmentary descriptions in the specification document.

Practices define required and recommended behaviours in prose or pseudo code. Practices are a critical component to interoperability because they establish expectations and processes for how MPV content is handled.

Modules are a grouping of Schema and Practices and are the unit of design that provides a coherent set of capabilities. Modules are indivisible; they cannot be subdivided. Modules may be combined if designed to be compatible.

Profiles are a set of Modules and are the unit of formal specification, of specification implementation and of specification compliance. Products can implement or not implement profiles. Each profile in MultiPhoto/Video defines only those modules that are necessary for the key tasks targeted by the profile.

Chapter 3: MPV Presentation Profile 1.0

The MPV Presentation Profile 1.0 is designed to accomplish the following key tasks: definition of albums and viewing a slideshow and interactively browsing the album. An album is a presentation-focused view of the collections of photo-video assets defined by the Basic Profile **[MPV-Basic]**.

The MPV Presentation Profile 1.0 consists of the following modules and practices, which are specified in detail separately in this document.

- MPV Core Module Schema 1.0
- MPV Core Module Practices 1.0
- MPV VXMP Module Schema 1.0
- MPV VXMP Module Practices 1.0
- MPV Presentation Module Schema 1.0
- MPV Presentation Module Practices 1.0

The MPV Presentation Profile is expected to be supported by most MPV-aware applications and devices that present collections to users and provides the basis for interoperability of collections across all range of storage media, devices, applications, and services.

Schema group	Namespace Identifier	Schema Location	Conventional Namespace Prefix	Specified Prefix for Namespace-Incompatible Environments
Presentation Profile	http://ns.osta.org/mpv/presentation/1.0/	profile/basic/basic.xsd	mpv:	mpv_

Chapter 4: MPV Presentation Module Schema

4.1 Module Introduction

The MPV Presentation Module define a few key elements and attributes. These are focused on enhancing the presentation experience. The key components are:

- Album – presentation of a set of assets
- Media Presentation Control Attributes – presentation-related information, such as duration
- Transition Filter – transition effects between assets

These changes are made by defining a new module. The module in part extends the MPV Core Module schema.

4.2 Schema Information

The MPV presentation module uses the mpvp: and mpvpTrans: namespace. The introductory schema information is expressed as follows.

Schema group	Namespace Identifier	Conventional Namespace Prefix	Specified Prefix for Namespace-Incompatible Environments
Presentation	http://ns.osta.org/mpv/presentation/1.0/	mpvp:	mpvp_
Presentation TransitionFilter	http://ns.osta.org/mpv/presentation/1.0/TransitionFilter	mpvpTrans:	mpvpTrans_
Presentation Defaults	http://ns.osta.org/mpv/presentation/1.0/Default	mpvpDflt:	mpvpDflt_

The presentation schema is defined using the following namespaces.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
xmlns:mpv="http://ns.osta.org/mpv/1.0/"
xmlns:mpvp="http://ns.osta.org/mpv/presentation/1.0/"
xmlns:mpvpTrans="http://ns.osta.org/mpv/presentation/1.0/TransitionFilter"
```



```
xmlns:mpvpDflt="http://ns.osta.org/mpv/presentation/1.0/Dflt"
xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="qualified">
```

4.3 <mpv:Manifest>

The MPV manifest groups all the MPV components into a single XML document. The Presentation Profile extends the definition of a MPV manifest to contains zero or more albums and an asset list.

In typical usage, a MPV manifest is stored in a stand-alone file; when it contains multiple albums, the first is the default. Any appliation that produces or consumes MPV content stored in stand-alone files in a storage filesystem shall be compliant with the Manifest schema and practices specification.

By implication of terminology, an MPV manifest contains reference to all the content that is relevant to a collection – it makes manifest the collection; it is a manifest of the collection.

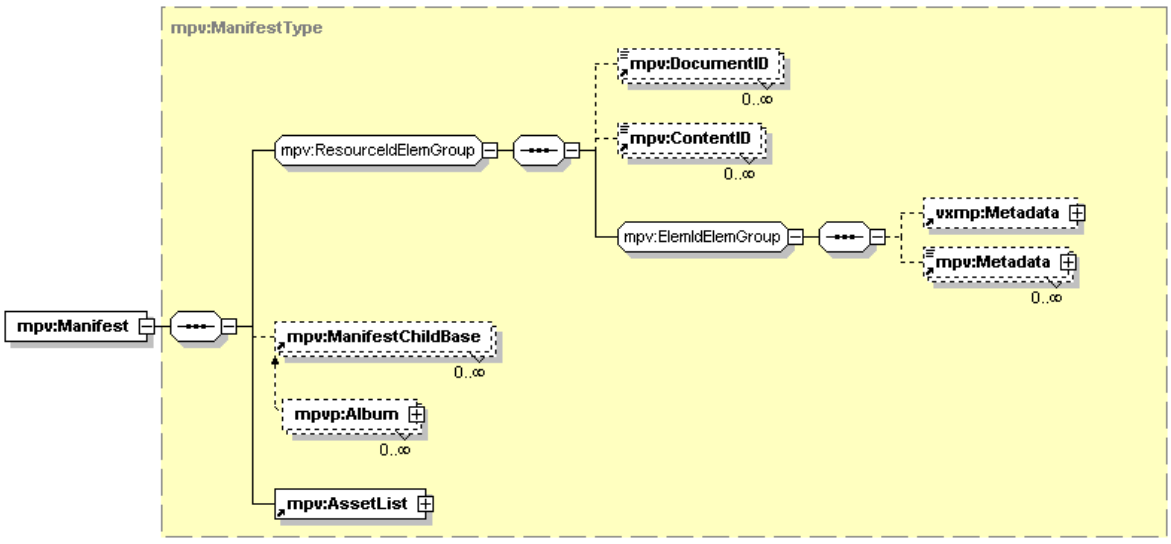
The top-level element of an MPV manifest is <mpv:Manifest>.

It may contain zero or mo re albums. The first album in the sequence is processed first and is the default album. A typical implementation will use this album to reference other albums, either in the same document or in other documents.

The mpv manifest always contains an asset list.

element **mpv:Manifest,** complexType **mpv:ManifestType**
element **mpv:Manifest**

diagram



namespace http://ns.osta.org/mpv/1.0/

type **mpv:ManifestType**

children **mpv:DocumentID** **mpv:ContentID** **Metadata** **mpv:Metadata** **mpv:ManifestChildBase** **mpv:AssetList**

source
```
<xs:element name="Manifest" type="mpv:ManifestType"/>
```

source
```
<xs:complexType name="ManifestType">
  <xs:sequence>
```



```
    <xs:group ref="mpv:ResourceIdElemGroup"/>
    <xs:element ref="mpv:ManifestChildBase" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="mpv:AssetList"/>
  </xs:sequence>
</xs:complexType>
```

4.4 <mpvp:Album>

An album defines a collection of media assets. They are organized in foreground and background collections. During playback, foreground and background are rendered in parallel for the slideshow. For interactive browsing, only foreground assets are used.

A typical use of the Background element is to specify a backdrop still image to underly the thumbnails during interactive browsing and a sequence of audios to provide music during the slideshow.

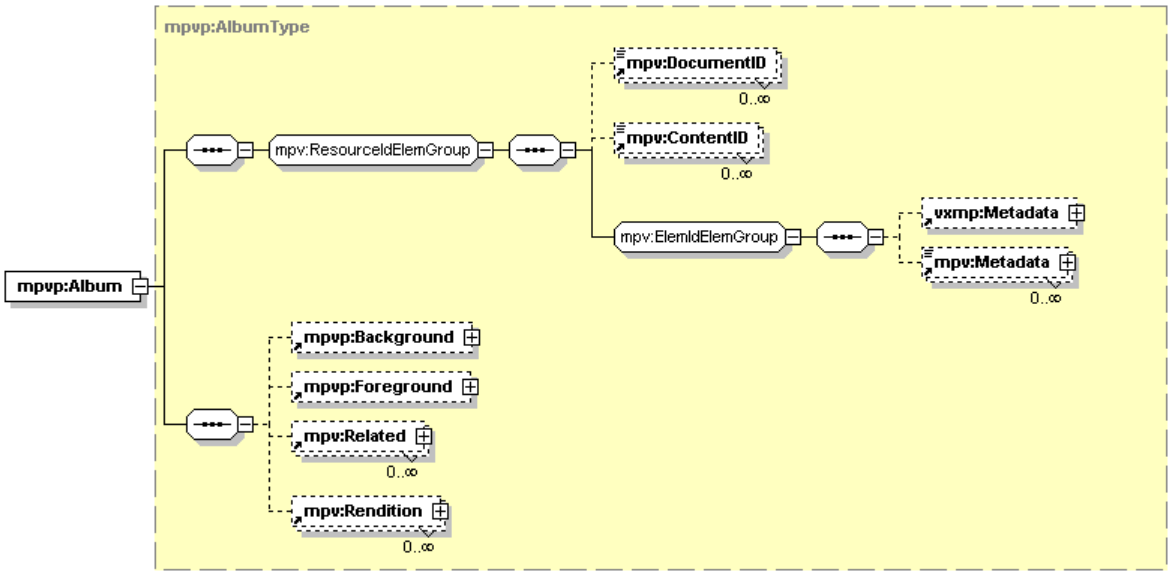
Arbitrary data can be attached to a collection, but it carries no explicit semantics with respect to MPV collection processing other than the data is associated with the collection. The Data element can make reference to data files associated with the Album but which are not media assets or are not Foreground or Background album items. For example, a file such as the DPOF AUTPRINT.MRK datafile placed in the DCF /MISC directory could be referenced using the Data element.

Renditions of an album represent derivatives of the collection. Typical renditions include a thumbnail representation of the album, a rendered video of the slideshow and print-formatted pages of the collection.

MarkLists of an album represent lists of album items that are distinguished in some manner. Two marklist types are defined for selected and hidden album items.

element **mpvp:Album**

diagram



namespace http://ns.osta.org/mpv/presentation/1.0/

type **mpvp:AlbumType**

children **mpv:DocumentID** **mpv:ContentID** **Metadata** **mpv:Metadata** **mpvp:Background** **mpvp:Foreground** **mpv:Related** **mpv:Rendition**

attributes

Name	Type	Use	Default	Fixed
mpv:id	xs:ID			
mpv:instanceID	xs:anyURI			
mpv:documentID	xs:anyURI			
mpv:contentID	xs:anyURI			

source

```
<xs:element name="Album" type="mpvp:AlbumType" substitutionGroup="mpv:ManifestChildBase"/>
```

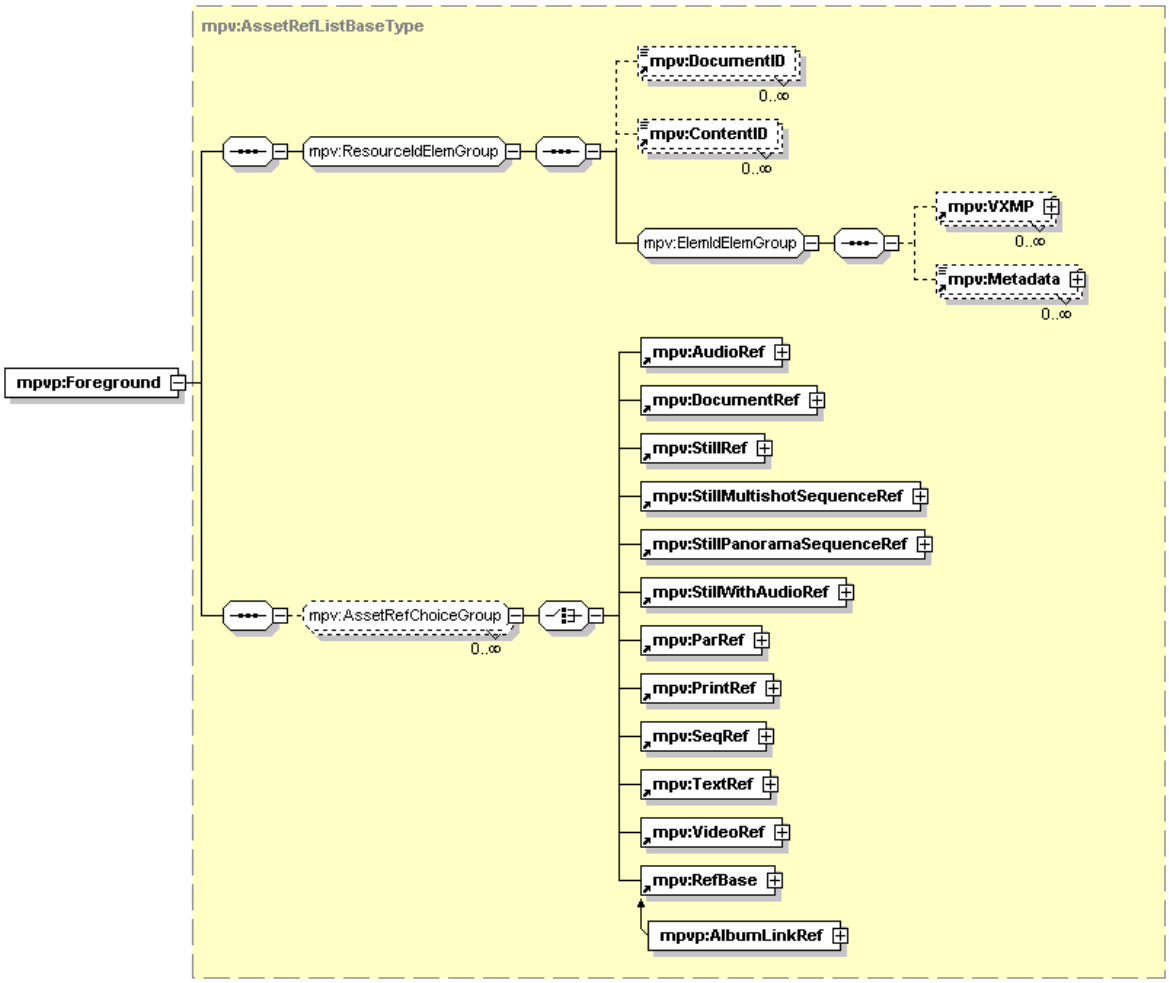
source

```
<xs:complexType name="AlbumType">
  <xs:complexContent>
    <xs:extension base="mpv:ManifestChildType">
      <xs:sequence>
        <xs:element ref="mpvp:Background" minOccurs="0"/>
        <xs:element ref="mpvp:Foreground" minOccurs="0"/>
        <xs:element ref="mpv:Related" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="mpv:Rendition" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

4.5 <mpvp:Foreground>, <mpvp:Background>

The MPV specification allows collections of assets to be organized conceptually into foreground and background content. The player decides how best to mix and present these contents. Additionally, assets may be organized into the generic RelatedDocuments group, which carries no specific semantics.

element **mpvp:Foreground**

diagram	
namespace	http://ns.osta.org/mpv/presentation/1.0/
type	**mpv:AssetRefListBaseType**
children	**mpv:DocumentID mpv:ContentID mpv:VXMP mpv:Metadata mpv:AudioRef mpv:DocumentRef mpv:StillRef mpv:StillMultishotSequenceRef mpv:StillPanoramaSequenceRef mpv:StillWithAudioRef mpv:ParRef mpv:PrintRef mpv:SeqRef mpv:TextRef mpv:VideoRef mpv:RefBase**
used by	complexType **mpvp:AlbumType**

attributes

Name	Type	Use	Default	Fixed
mpv:id	xs:id			
mpv:instanceID	xs:anyURI			
mpv:documentID	xs:anyURI			
mpv:contentID	xs:anyURI			
defaultListIDRef	xs:idref	optional		

source

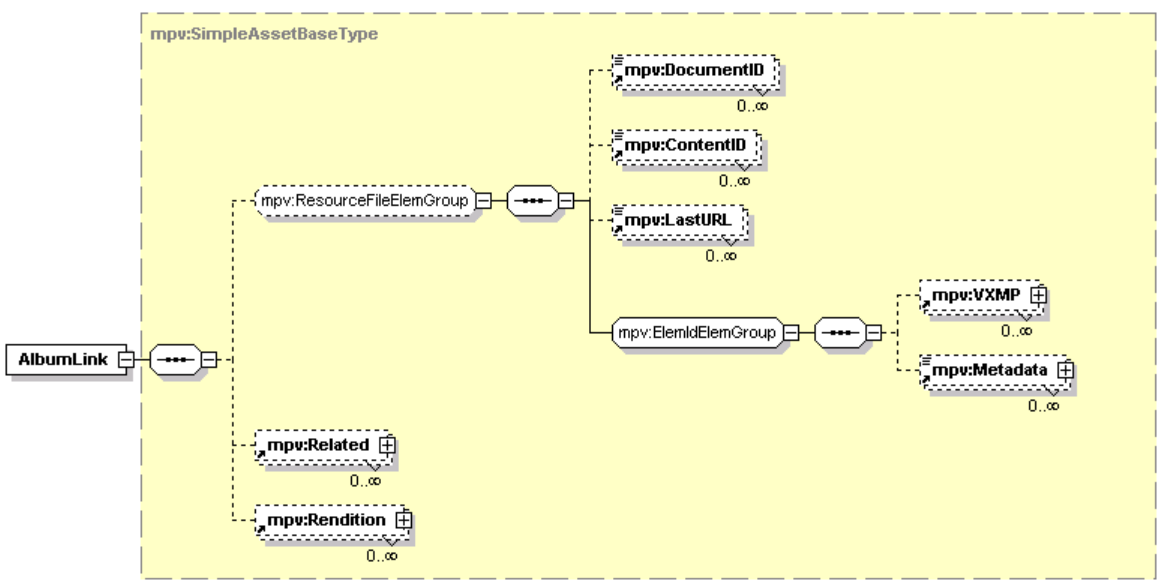
```
<xs:element name="Foreground" type="mpv:AssetRefListBaseType"/>
```

The Background element diagram is identical.

4.6 <mpvp:AlbumLink>

An AlbumLink element references another album using the same identifiers as all other media assets. Renditions may be supplied. A typical rendition would be a thumbnail representing the album.

element **AlbumLink**

diagram	
namespace	http://ns.osta.org/mpv/presentation/1.0/
type	mpv:SimpleAssetBaseType
children	mpv:DocumentID mpv:ContentID mpv:LastURL mpv:VXMP mpv:Metadata mpv:Related mpv:Rendition

attributes

Name	Type	Use	Default	Fixed
mpv:id	xs:id			
mpv:instanceID	xs:anyURI			
mpv:documentID	xs:anyURI			
mpv:contentID	xs:anyURI			
mpv:lastURL	xs:anyURI			
mpv:byteOffset	xs:integer			
mpv:xmlPacket	xs:integer			
mpv:leaseExpiresDate	xs:date			
mpv:leaseDur	xs:float			
mpv:leaseID	xs:string			

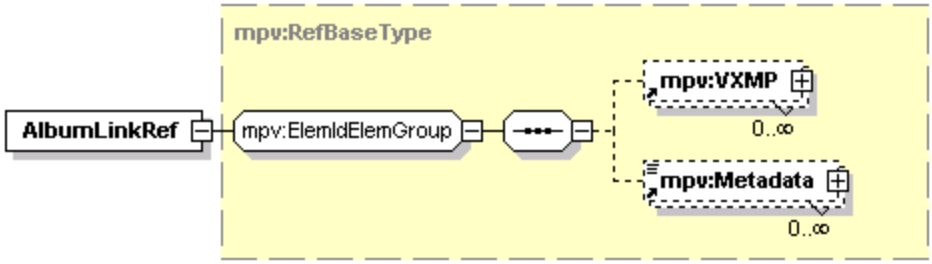
source

```
<xs:element name="AlbumLink" type="mpv:SimpleAssetBaseType" " substitutionGroup="mpv:SimpleAssetBase"/>
```

4.7 <mpvp:AlbumLinkRef>

An AlbumLinkRef element references an AlbumLink asset

element **AlbumLinkRef**

diagram	
namespace	http://ns.osta.org/mpv/presentation/1.0/
type	**mpv:RefBaseType**
children	**mpv:VXMP** **mpv:Metadata**
attributes	Name / Type / Use / Default / Fixed listIDRef / xs:idref / optional idRef / xs:idref / required mpv:id / xs:id
source	<xs:element name="AlbumLinkRef" type="mpv:RefBaseType" substitutionGroup="mpv:RefBase" />

4.8 <mpvpCtrl:Properties> Media Asset Presentation Control

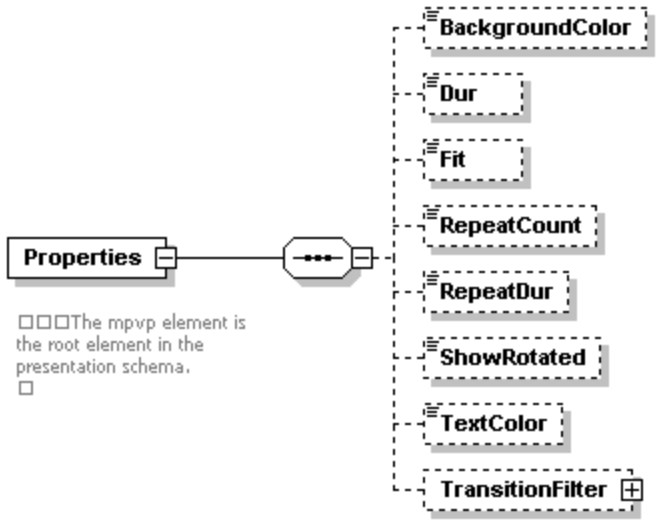
The Presentation Profile defines a schema for presentation properties. This schema can be used on all media assets by specying the root element of the mpvp schema as the only child of the vxmp :Metadata element.

This schema is derived from the SMIL specifications [SMIL10] and [SMIL20]. The SMIL language and SMIL players are seen as an attractive tools for presenting MPV documents. A carefully constrained set of SMIL elements and attributes are chosen for this basic presentation schema. The emphasis was on selecting just those features essential to deliver a basic user experience that is easy to use and compelling and that can also be implementated across many platforms and embedded devices.

The guiding practice for applications and devices that process and present MPV content based on this schema is that presentation properties inherit to lower areas of scope. For example, a <mpvp:Fit> value specified as metadata of the <mpv:Foreground> element itself will apply to all its children.

element **Properties**

diagram



namespace http://ns.osta.org/mpv/presentation/1.0/Control

type extension of **PropertiesType**

children **BackgroundColor** **Dur** **Fit** **RepeatCount** **RepeatDur** **ShowRotated** **TextColor** **TransitionFilter**

annotation documentation

The mpvp element is the root element in the presentation schema.

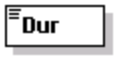
source

```
<xs:element name="Properties" substitutionGroup="vxmp:Properties">
 <xs:annotation>
  <xs:documentation>
                                The mpvp element is the root element in the presentation schema.
          </xs:documentation>
 </xs:annotation>
 <xs:complexType>
  <xs:complexContent>
   <xs:extension base="vxmp:PropertiesType">
    <xs:sequence>
     <xs:element name="BackgroundColor" type="xs:string" minOccurs="0"/>
     <xs:element name="Dur" type="mpvpCtrl:DurationType" minOccurs="0"/>
     <xs:element name="Fit" type="mpvpCtrl:FitType" minOccurs="0"/>
     <xs:element name="RepeatCount" type="xs:string" minOccurs="0"/>
     <xs:element name="RepeatDur" type="xs:string" minOccurs="0"/>
     <xs:element name="ShowRotated" type="xs:integer" minOccurs="0"/>
     <xs:element name="TextColor" type="xs:string" minOccurs="0"/>
     <xs:element name="TransitionFilter" minOccurs="0">
      <xs:complexType>
       <xs:sequence>
        <xs:element ref="mpvpTrans:StructBase" minOccurs="0"/>
       </xs:sequence>
      </xs:complexType>
     </xs:element>
    </xs:sequence>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
</xs:element>
```


4.8.1 Properties: mpvpCtrl:Dur

Specifies the simple duration.

element **Properties/Dur**

diagram	
namespace	http://ns.osta.org/mpv/presentation/1.0/Control
type	**DurationType**
source	<xs:element name="Dur" type="mpvpCtrl:DurationType" minOccurs="0"/>

simpleType **DurationType**

namespace	http://ns.osta.org/mpv/presentation/1.0/Control
type	**xs:string**
used by	element **Properties/Dur**
source	<xs:simpleType name="DurationType"> <xs:restriction base="xs:string"/> </xs:simpleType>

The attribute value can be any of the following:

Clock-value

Specifies the length of the simple duration, measured in element active time. Value must be greater than 0.

Clock values have the following syntax:

```
Clock-value          ::= Timecount-value
Timecount-value      ::= Timecount ("." Fraction)?
Fraction             ::= DIGIT+
Timecount            ::= DIGIT+
DIGIT                ::= [0-9]
```

"media"

Specifies the simple duration as the intrinsic media duration. This is only valid for elements that define media.

"indefinite"

Specifies the simple duration as indefinite.

4.8.2 Properties: mpvpCtrl:RepeatCount

Specifies the number of iterations of the simple duration.

element **Properties/RepeatCount**

diagram	

namespace	http://ns.osta.org/mpv/presentation/1.0/Control
type	**xs:string**
source	<xs:element name="RepeatCount" type="xs:string" minOccurs="0"/>

It can have the following attribute values:

numeric value

This is a (base 10) "floating point" numeric value that specifies the number of iterations. It can include partial iterations expressed as fraction values. A fractional value describes a portion of the simple duration. Values must be greater than 0.

"indefinite"

The element is defined to repeat indefinitely (subject to the constraints of the parent time container).

4.8.3 Properties: mpvpCtrl:RepeatDur

Specifies the total duration for repeat.

element **Properties /RepeatDur**

diagram	
namespace	http://ns.osta.org/mpv/presentation/1.0/Control
type	**xs:string**
source	<xs:element name="RepeatDur" type="xs:string" minOccurs="0"/>

It can have the following attribute values:

Clock-value

Specifies the duration in element active time to repeat the simple duration.

"indefinite"

The element is defined to repeat indefinitely (subject to the constraints of the parent time container).

4.8.4 Properties: mpvpCtrl:ShowRotated

Specifies the total degrees from 0 to 360 for rotation of the asset when presented.

element **Properties /ShowRotated**

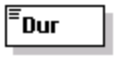
diagram	
namespace	http://ns.osta.org/mpv/presentation/1.0/Control
type	**xs:integer**
source	<xs:element name="ShowRotated" type="xs:integer" minOccurs="0"/>

The default value is 0.

4.8.5 Properties: mpvpCtrl:Fit

This attribute specifies the behavior if the intrinsic height and width of a visual media asset differ from the values specified by the height and width attributes in the "region" element. This attribute replaces the behavior defined in CSS2.

element **Properties /Fit**

diagram	
namespace	http://ns.osta.org/mpv/presentation/1.0/Control
type	**FitType**
source	<xs:element name="Fit" type="mpvpCtrl:FitType" minOccurs="0"/>

simpleType **FitType**

namespace	http://ns.osta.org/mpv/presentation/1.0/Control
type	union of (**FitBaseType** , **xs:anyURI**)
used by	element **Properties/Fit**
source	<xs:simpleType name="FitType"> <xs:union memberTypes="mpvpCtrl:FitBaseType xs:anyURI"/> </xs:simpleType>

simpleType **FitBaseType**

namespace	http://ns.osta.org/mpv/presentation/1.0/Control
type	restriction of **xs:string**
used by	simpleType **FitType**
facets	enumeration hidden enumeration fill enumeration meet enumeration scroll enumeration slide
source	<xs:simpleType name="FitBaseType"> <xs:restriction base="xs:string"> <xs:enumeration value="hidden"/> <xs:enumeration value="fill"/> <xs:enumeration value="meet"/> <xs:enumeration value="scroll"/> <xs:enumeration value="slide"/> </xs:restriction> </xs:simpleType>

This attribute can have the following values:

fill

Scale the asset's height and width independently so that the content just touches all edges of the box.

hidden (default)

- If the intrinsic height (width) of the media asset element is smaller than the height (width) defined in the "region" element, render the asset starting from the top (left) edge and fill up the remaining height (width) with the background color.
- If the intrinsic height (width) of the media asset element is greater than the height (width) defined in the "region" element, render the asset starting from the top (left) edge until the height (width)

defined in the "region" element is reached, and clip the parts of the asset below (right of) the height (width).

meet

Scale the visual media asset while preserving its aspect ratio until its height or width is equal to the value specified by the height or width attributes, while none of the content is clipped. The asset's left top corner is positioned at the top-left coordinates of the box, and empty space at the left or bottom is filled up with the background color.

scroll

A scrolling mechanism should be invoked when the element's rendered contents exceed its bounds.

slice

Scale the visual media asset while preserving its aspect ratio so that its height or width are equal to the value specified by the height and width attributes while some of the content may get clipped. Depending on the exact situation, either a horizontal or a vertical slice of the visual media asset is displayed. Overflow width is clipped from the right of the media asset. Overflow height is clipped from the bottom of the media asset.

4.8.6 Properties: mpvpCtrl:BackgroundColor

Specifies the background color of the element and all subelements. This is used to fill any visual region not occluded by the asset's rendition. The default background color is "transparent", which implies that, by default, the implementation dependent window background will be shown. The vocabulary for BackgroundColor is defined by CSS2 system colors **[CSS2]**.

element **Properties /BackgroundColor**

diagram	
namespace	http://ns.osta.org/mpv/presentation/1.0/Control
type	**xs:string**
source	<xs:element name="BackgroundColor" type="xs:string" minOccurs="0"/>

4.8.7 Properties: mpvpCtrl:TextColor

Specifies the text color of the element and all subelements. The default text color is implemenation dependent, which implies that, by default, the implementation dependent window background color will be shown under the implementation dependent text color. The vocabulary for TextColor is defined by CSS2 system colors **[CSS2]**.

element **Properties /TextColor**

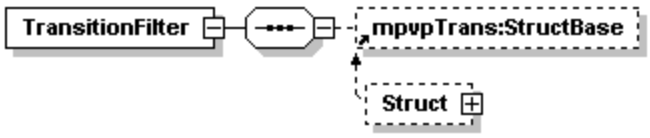
diagram	
namespace	http://ns.osta.org/mpv/presentation/1.0/Control
type	**xs:string**
source	<xs:element name="TextColor" type="xs:string" minOccurs="0"/>

4.8.8 Property: mpvpCtrl:TransitionFilter

Specifies the transitionFilter that should be applied to the media asset.

element **Properties/TransitionFilter**

diagram



namespace http://ns.osta.org/mpv/presentation/1.0/Control

children **StructBase**

source

```
<xs:element name="TransitionFilter" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="mpvpTrans:StructBase" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

The value of the property is an mpvpTrans:mpvpTrans element (see below).

4.8.9 Example of syntax

The mpvp schema has a root element, mpvp:mpvp that appears the only child of the xmp:XMP metadata element. In the following example, only two of the optional properties of the schema are specified for the Still, the backgroundColor and the Fit properties.

```
<mpv:Manifest
     xmlns:mpv=”http://ns.osta.org/mpv/1.0/”
     xmlns:vxmp=”http://ns.vxmp.org/vxmp/1.0”
     xmlns:mpvpCtrl=”http://ns.osta.org/mpv/presentation/1.0/mpvpCtrl”
     xmlns:mpvpTrans=”http://ns.osta.org/mpv/presentation/1.0/mpvpTrans”
>
  . . .
  <mpv:Still>
     . . .
     <vxmp:Metadata>
            <Properties xmlns=”http://ns.osta.org/mpv/presentation/1.0/mpvpCtrl”>
                <BackgroundColor>Blue</mpvpCtrl:BackgroundColor>
                <TextColor>White</mpvpCtrl:TextColor>
                <Fit>meet</mpvpCtrl:Fit>
                <TransitionFilter>
                    <Struct xmlns=”http://ns.osta.org/mpv/presentation/1.0/mpvpTrans”>
                        <Type>barWipe</Type>
                    </Struct>
                </TransitionFilter>
            </Properties>
     </vxmp:Metadata>
  </mpv:Still>
  . . .
```



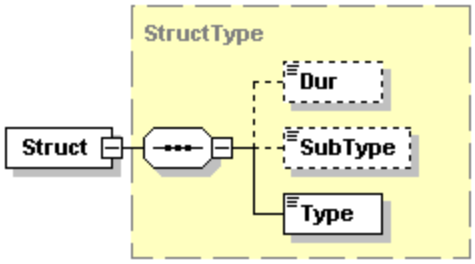
```
</mpv:Manifest>
```

4.9 <mpvpTrans:StructBase> Transition Filter

A transition filter that implements a transition from the asset before to the asset after. It is applied at the completion of presenting the asset on which it is defined and transitioning into the next asset that is defined. This element is strictly presentation oriented. It is specified as the value of the TransitionFilter property in the mpvp schema.

element **Struct**

diagram



namespace http://ns.osta.org/mpv/presentation/1.0/TransitionFilter

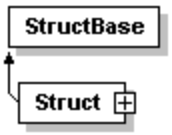
type **StructType**

children **Dur** **SubType** **Type**

source
```
<xs:element name="Struct" type="StructType" substitutionGroup="StructBase"/>
```

element **StructBase**

diagram



namespace http://ns.osta.org/mpv/presentation/1.0/TransitionFilter

type **StructBaseType**

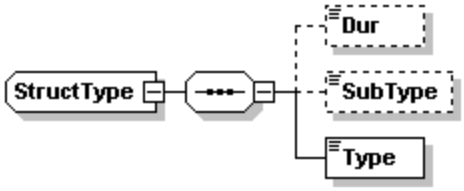
used by element **Properties/TransitionFilter**

source
```
<xs:element name="StructBase" type="StructBaseType" abstract="true"/>
```

source
```
<xs:complexType name="StructBaseType">
  <xs:complexContent>
    <xs:extension base="vxmp:StructType"/>
  </xs:complexContent>
</xs:complexType>
```


complexType **StructType**

diagram



namespace http://ns.osta.org/mpv/presentation/1.0/TransitionFilter

type extension of **StructBaseType**

children **Dur** **SubType** **Type**

used by element **Struct**

source

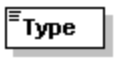
```
<xs:complexType name="StructType">
  <xs:complexContent>
    <xs:extension base="StructBaseType">
      <xs:sequence>
        <xs:element name="Dur" type="xs:float" minOccurs="0"/>
        <xs:element name="SubType" type="xs:string" minOccurs="0"/>
        <xs:element name="Type" type="xs:string"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

4.9.1 Property: mpvpTrans:Type

This is the type or family of transition. This attribute is required and must be one of the transition families listed.

element **StructType/Type**

diagram



namespace http://ns.osta.org/mpv/presentation/1.0/TransitionFilter

type **xs:string**

source

```
<xs:element name="Type" type="xs:string"/>
```

4.9.2 Property: mpvpTrans:Subtype

This is the subtype of the transition. This parameter is optional and if specified, must be one of the transition subtypes appropriate for the specified type as listed. If this parameter is not specified then the transition reverts to the default subtype for the specified transition type.

element **StructType/SubType**

diagram

namespace http://ns.osta.org/mpv/presentation/1.0/TransitionFilter

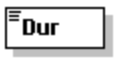
type **xs:string**

source <xs:element name="SubType" type="xs:string" minOccurs="0"/>

4.9.3 Property: mpvpTrans:Dur

The default duration is the intrinsic duration built into the transition. All of the transitions have a default duration of 1 second.

element **StructType/Dur**

diagram 

namespace http://ns.osta.org/mpv/presentation/1.0/TransitionFilter

type **xs:float**

source <xs:element name="Dur" type="xs:float" minOccurs="0"/>

4.9.4 Recommended Transitions

MPV Player implementations are not required to implement any transitions. If they do implement transitions, the following transitions types are recommended to be implemented first.

Transition type	**Default Transition subtype**
barWipe	leftToRight
irisWipe	rectangle
clockWipe	clockwiseTwelve
snakeWipe	topLeftHorizontal

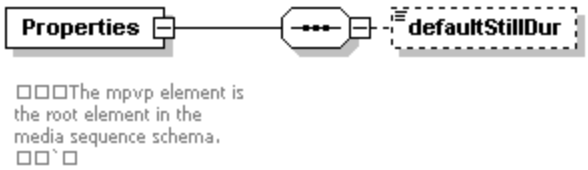
Please refer to the appendix for a complete set of defined transition types and subtypes.

4.10 <mpvp:Default>

The Default element specifies default presentation parameters for presentation. Typical usage will be on the top-level Album or on Foreground or Background elements.

element **Properties**

diagram



namespace http://ns.osta.org/mpv/presentation/1.0/Sequence

type extension of **PropertiesType**

children **defaultStillDur**

annotation documentation

The mpvp element is the root element in the media sequence schema.

source

```
<xs:element name="Properties" substitutionGroup="vxmp:Properties">
 <xs:annotation>
  <xs:documentation>
                The mpvp element is the root element in the media sequence schema.
                </xs:documentation>
 </xs:annotation>
 <xs:complexType>
  <xs:complexContent>
   <xs:extension base="vxmp:PropertiesType">
    <xs:sequence>
     <xs:element name="defaultStillDur" type="xs:float" minOccurs="0"/>
    </xs:sequence>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
</xs:element>
```

PROPERTY DESCRIPTIONS

4.10.1 Property: mpvpDflt:StillDur

Sets the duration for presenting a still image.

element **Properties/defaultStillDur**

diagram

namespace http://ns.osta.org/mpv/presentation/1.0/Sequence

type **xs:float**

source

```
<xs:element name="defaultStillDur" type="xs:float" minOccurs="0"/>
```


Chapter 5: MPV Presentation Module Practices

The Presentation Module practices are largely recommended because there is an on-going growth of basic and innovative presentation techniques and attributes. Rather than strive for completeness or overspecification of behaviour, MPV strives to provide just enough presentation schema and practices to enable applications to provide the user two basic capabilities: browsing and watching. Close behind in user value for photo-video content is printing. Additional presentation capabilities may be expressed as custom metadata and processed by specialized players.

5.1 Best Practices for Presenting a Manifest

A manifest may contain more than one Album. For presentation purposes, the first album is the default Album; other content may be ignored. If additional Albums in a manifest are to be presented, they must be referred to explicitly from another Album, such as the default Album; alternately, the presentation application may choose to present the user with the list of albums defined in the manifest.

A key presentation characteristic is perceived startup time from storage media insertion or presentation invocation to begin of the presentation. This performance can be accelerated in various ways:

- place the manifest in a location that is quickly found by the scanning algorithm for manifests
- provide the user positive feedback that the Manifest is being loaded or processed
- load only one album at a time
- provide Screen and Thumbnail Renditions for images and video
- present using placeholders for assets whose lastURL values require fixup; then slowly fixup broken lastURLs

5.2 Best Practices for Watching

The basic watch experience is to play back the content in the MPV album as follows. The Background and Foreground parts of an Album have the same expressive power. Background assets and metadata are played under foreground assets and metadata, both in z-order and in audio mixing.

Using this behavior, a typical watch experience of a slideshow can be provided:

- Foreground seqence of Still, Video, and Audio content
- Background music track of Audio content

For better performance, use of "Screen" Renditions is recommended when present.

More advanced watching applications will provide the user to pause, rewind, and fast forward through the content. Additional operations may be available directly on the content as watched, such as "Mark" or "Print".

UNSUPPORTED TYPES

When an asset has a media type that isn't supported for playback (e.g. a GIF image), the recommended behaviour for the watch application is to skip silently over the unsupported asset.

ALBUMREF

When an asset is an AlbumRef, the recommended behaviour for the watch application is to skip silently over the AlbumRef.

PROVIDED SHOWS

Watch applications should study the Renditions available on the Album or Foreground for a "Show" Rendition video. This represents a pre-computed rendition of the watch experience and should be used in preference to the watch experience the application can produce itself or at least presented as an option to the user. The "Show" rendition provides a means for sophisticated authoring and production applications to separately produce an advanced watch experience that can be accessed simply by playing a video. There is a user expectation that the Show content and Album content be approximately the same.

5.3 Best Practices for Browsing

The basic browsing experience should provide two basic capabilities:

- browsing of thumbnails of photo-video content
- browsing of full-screen views of the photo-video content

For better performance in thumbnail mode, use of "Thumbnail" Renditions for both stills and video is recommended. For better performance in full-screen mode, use of "Screen" Renditions for stills is recommended.

Advanced browsing applications will treat StillsMultishotSequence and StillsPanoramaSequence specially. For example, in thumbnail browse mode, they may show just one of the stills and iconically represent that the item is a sequence of stills.

UNSUPPORTED TYPES

When an asset has a media type that isn't supported for playback (e.g. a GIF image), the recommended behaviour for the browsing application is to show an empty thumbnail with a message that the media type is not supported by this player.

ALBUMREFS

When the Album contains AlbumRef, these are presented to the user as choices to link to another Album. A "Thumbnail" or "Screen" Rendition on the AlbumRef can offer a visual depiction of the target Album.

When browsing, it is recommended that AlbumRefs be presented just as another type of asset, albeit one that when selected opens another album.

RENDITIONS

Other renditions at the album and asset levels may be of interest. In particular, Print assets may offer useful content to print, such as thumbnail index pages or CD labels. Browsing applications are recommended to provide the user access to renditions and assets of the type "Print" and "Text".

5.4 Best Practices for Supported Formats

The following formats are recommended to be supported for playback by players that desire to play a large percentage of the photo-video content they may encounter.

Stills:

- JPEG, in both Exif and JFIF variations
- TIFF
- GIF
- PNG

Video:

- AVI MJPEG
- MOV PJPEG
- MPEG1
- MPEG1 VideoCD
- MPEG2

Audio:

- WAV
- MPEG1 Layer 3 (MP3)
- MPEG1 Layer 2

Print:

- PDF

5.5 Examples

[TODO]

5.5.1 Startup List of Albums with Background Image

5.5.2 Representing a Title Image

5.5.3 Album Renditions of a Video Slideshow and Printed Content

5.5.4 Building Up a StillSequenceWithAudio Type

Appendix I: Transition Types Reference

The following table is excerpted from the SMIL 2.0 specification. It lists the vocabulary of defined transition types and subtypes. The SMPTE Wipe Codes (where appropriate) are provided in parentheses after the subtype name and are for reference only. The Wipe Codes are not part of the transition subtype name. The default transition subtype for each type is indicated by the word [default].

Transition type	**Transition subtypes** (SMPTE Wipe Codes in parentheses)
Edge Wipes - wipes occur along an edge	
"barWipe"	"leftToRight" (1) [default], "topToBottom" (2)
"boxWipe"	"topLeft" (3) [default], "topRight" (4), "bottomRight" (5), "bottomLeft" (6), "topCenter" (23), "rightCenter" (24), "bottomCenter" (25), "leftCenter" (26)
"fourBoxWipe"	"cornersIn" (7) [default], "cornersOut" (8)
"barnDoorWipe"	"vertical" (21) [default], "horizontal" (22), "diagonalBottomLeft" (45), "diagonalTopLeft" (46)
"diagonalWipe"	"topLeft" (41) [default], "topRight" (42)
"bowTieWipe"	"vertical" (43) [default], "horizontal" (44)
"miscDiagonalWipe"	"doubleBarnDoor" (47) [default], "doubleDiamond" (48)
"veeWipe"	"down" (61) [default], "left" (62), "up" (63), "right" (64)
"barnVeeWipe"	"down" (65) [default], "left" (66), "up" (67), "right" (68)
"zigZagWipe"	"leftToRight" (71) [default], "topToBottom" (72)
"barnZigZagWipe"	"vertical" (73) [default], "horizontal" (74)
Iris Wipes - shapes expand from the center of the media	
"irisWipe"	"rectangle" (101) [default], "diamond" (102)
"triangleWipe"	"up" (103) [default], "right" (104), "down" (105), "left" (106)

"arrowHeadWipe"	"up" (107) [default], "right" (108), "down" (109), "left" (110)
"pentagonWipe"	"up" (111) [default], "down" (112)
"hexagonWipe"	"horizontal" (113) [default], "vertical" (114)
"ellipseWipe"	"circle" (119) [default], "horizontal" (120), "vertical" (121)
"eyeWipe"	"horizontal" (122) [default], "vertical" (123)
"roundRectWipe"	"horizontal" (124) [default], "vertical" (125)
"starWipe"	"fourPoint" (127) [default], "fivePoint" (128), "sixPoint" (129)
"miscShapeWipe"	"heart" (130) [default], "keyhole" (131)
Clock Wipes - rotate around a center point	
"clockWipe"	"clockwiseTwelve" (201) [default], "clockwiseThree" (202), "clockwiseSix" (203), "clockwiseNine" (204)
"pinWheelWipe"	"twoBladeVertical" (205) [default], "twoBladeHorizontal" (206), "fourBlade" (207)
"singleSweepWipe"	"clockwiseTop" (221) [default], "clockwiseRight" (222), "clockwiseBottom" (223), "clockwiseLeft" (224), "clockwiseTopLeft" (241), "counterClockwiseBottomLeft" (242), "clockwiseBottomRight" (243), "counterClockwiseTopRight" (244)
"fanWipe"	"centerTop" (211) [default], "centerRight" (212), "top" (231), "right" (232), "bottom" (233), "left" (234)
"doubleFanWipe"	"fanOutVertical" (213) [default], "fanOutHorizontal" (214), "fanInVertical" (235), "fanInHorizontal" (236)
"doubleSweepWipe"	"parallelVertical" (225) [default], "parallelDiagonal" (226), "oppositeVertical" (227), "oppositeHorizontal" (228), "parallelDiagonalTopLeft" (245), "parallelDiagonalBottomLeft" (246)
"saloonDoorWipe"	"top" (251) [default], "left" (252), "bottom" (253), "right" (254)
"windshieldWipe"	"right" (261) [default], "up" (262), "vertical" (263), "horizontal" (264)
Matrix Wipes - media is revealed in squares following a pattern	
"snakeWipe"	"topLeftHorizontal" (301) [default], "topLeftVertical" (302), "topLeftDiagonal" (303), "topRightDiagonal" (304), "bottomRightDiagonal" (305), "bottomLeftDiagonal" (306)
"spiralWipe"	"topLeftClockwise" (310) [default], "topRightClockwise" (311), "bottomRightClockwise" (312), "bottomLeftClockwise" (313), "topLeftCounterClockwise" (314), "topRightCounterClockwise" (315), "bottomRightCounterClockwise" (316), "bottomLeftCounterClockwise" (317)
"parallelSnakesWipe"	"verticalTopSame" (320), [default] "verticalBottomSame" (321), "verticalTopLeftOpposite" (322), "verticalBottomLeftOpposite" (323), "horizontalLeftSame" (324), "horizontalRightSame" (325), "horizontalTopLeftOpposite" (326), "horizontalTopRightOpposite" (327), "diagonalBottomLeftOpposite" (328), "diagonalTopLeftOpposite" (329)
"boxSnakesWipe"	"twoBoxTop" (340) [default], "twoBoxBottom" (341), "twoBoxLeft" (342), "twoBoxRight" (343), "fourBoxVertical (344), "fourBoxHorizontal" (345)

"waterfallWipe"	"verticalLeft" (350) [default], "verticalRight" (351), "horizontalLeft" (352), "horizontalRight" (353)
Non-SMPTE Wipes	
"pushWipe"	"fromLeft" [default], "fromTop", "fromRight", "fromBottom"
"slideWipe"	"fromLeft" [default], "fromTop", "fromRight", "fromBottom"
"fade"	"crossfade" [default], "fadeToColor", "fadeFromColor"

The "pushWipe" transitions looks as if the destination media "pushes" the background media away. In other words, both the background media and the destination media are moving.

In the "slideWipe" transitions, the destination media moves, but the background media does not. The visual effect of "slideWipe" transitions is that the destination media is "sliding" in across the background media.

The "fade" transitions are pixel-by-pixel blends between the destination media and either the background media or a specified color.

Appendix II: Typographic Conventions

[TODO] – this is out of date and laxly followed.

Schema are in yellow boxes in Courier font.

```
Plain text is schema.
```

Examples of MPV metadata structures are in Courier font.

```
Example.

<MPV>
   <ALBUM>
      ...
   </ALBUM>
</MPV>
```


Appendix III: References

[CSS2]

"Cascading Style Sheets, level 2", Bert Bos, Håkon Wium Lie, Chris Lilley, Ian Jacobs. W3C Recommendation 12 May 1998.
Available at http://www.w3.org/TR/REC-CSS2

[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/

[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

[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

[MPV-Basic]

"MultiPhoto/Video – Basic Profile Specification", June, 2002, Optical Storage Technology Association.
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

[SMIL10]

"Synchronized Multimedia Integration Language (SMIL) 1.0" P. Hoschka. W3C Recommendation 15 June 1998,
Available at http://www.w3.org/TR/REC-smil.

[SMIL20]

"Synchronized Multimedia Integration Language (SMIL 2.0) Specification". W3C Working Draft, work in progress.
Available at http://www.w3.org/TR/smil20/

[SMIL-MOD]

"Synchronized Multimedia Modules based upon SMIL 1.0", Patrick Schmitz, Ted Wugofski and Warner ten Kate. W3C Note 23 February 1999,
Available at http://www.w3.org/TR/NOTE-SYMM-modules

[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.

[VXMP]

"V*XMP – Validatable Extensible Metadata Platform – 17 June 2002"* , Copyright 2002 Hewlett-Packard Co.,
Available at …. [TODO – fixup]

[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

[XMP-FW]

"*XMP – Extensible Metadata Platform 14 Sept 01"* , Copyright 2001 Adobe Inc,
Available at http://xml.coverpages.org/XMP-MetadataFramework.pdf. Also at
http://partners.adobe.com/asn/developer/xmp/download/docs/MetadataFramework.pdf

[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/

