



# MultiPhoto/Video

*Manifest, Metadata and Practices for Digital Photo-Video Collections*



## Presentation Profile Specification

**Revision 0.40  
Working Draft**

**June 7, 2002**

**© 2001-2002 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 DISBTIBUTION ON A PUBLIC WEBSITE**

This document is available at <http://www.osta.org/mpv/mpvmb/rs/specs/MPV-Pres-Profile-Spec-0.40WD.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><a href="http://www.osta.org">http://www.osta.org</a></p> <p><u>MultiPhoto/Video Website</u> <a href="http://www.osta.org/mpv/">http://www.osta.org/mpv/</a></p>	<p><u>Technical Content</u></p> <p>Pieter van Zee Editor, MultiPhoto/Video 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@oaktech.com</p>
--	--

### 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

Copyright 2001-2002 Optical Storage Technology Association, Inc. All Rights Reserved.

## **LICENSING IMPORTANT NOTICES**

(a) This document is an authorized and approved publication of the Optical Storage Technology Association (OSTA). The specifications 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 and appropriate reference is made to the origin of the material. 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 unauthorized and specifically prohibited.

(b) 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, copyright, mask work right or trade secret right). By publication of this document, no position is taken by OSTA with respect to the validity or infringement of any patent 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) This document is a specification adopted by Optical Storage Technology Association (OSTA). This document may be revised by OSTA and users are advised to obtain the latest version. It is intended solely as a guide for companies interested in developing products which can be compatible with other products developed using this document. This document is provided "AS IS". OSTA makes no representation or warranty regarding this document, and anyone using this document shall do so at their 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 direct, indirect, exemplary, incidental, proximate or consequential damages or expenses arising from the use of this document for any reason whatsoever, even if OSTA is advised of a particular use of this document. This document defines only one approach to compatibility, and other approaches may be available in the industry.

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

# Contents

---

Contents .....	4
Chapter 1: Introduction.....	6
1.1 Executive Summary .....	6
1.2 Overview .....	7
1.3 Terms of Use.....	8
Chapter 2: Key Concepts of the MPV and Related Specifications .....	9
2.1 MPV Specification Architecture .....	9
2.2 Profiles and Modules, Schema and Practices .....	10
2.3 MPV Core and the MPV Basic and Presentation Profiles .....	10
2.3.1 MPV Core .....	10
2.3.2 MPV Basic Profile .....	10
2.3.3 MPV Presentation Profile .....	11
2.4 XML Usage.....	12
Chapter 3: MPV Presentation Profile 1.0 .....	13
Chapter 4: MPV Presentation Schemas .....	14
4.1 Introduction.....	14
4.2 Schema Information.....	14
4.3 <mpvp:Album> .....	14
4.4 <mpvp:Foreground>, <mpvp:Background>.....	16
4.5 <mpvp:AlbumRef> .....	18
4.6 <mpvpCtrl:ControlProperties> Media Asset Presentation Control.....	18
4.6.1 Properties: mpvpCtrl:BackgroundColor .....	19
4.6.2 Properties: mpvpCtrl:Dur .....	20
4.6.3 Properties: mpvpCtrl:Fit .....	21
4.6.4 Properties: mpvpCtrl:Margin .....	22
4.6.5 Properties: mpvpCtrl:RepeatCount .....	23
4.6.6 Properties: mpvpCtrl:RepeatDur .....	23
4.6.7 Properties: mpvpCtrl:ShowRotated .....	24
4.6.8 Properties: mpvpCtrl:StillDur .....	24
4.6.9 Property: mpvpCtrl:TransitionFilter.....	24
4.6.10 Example of syntax.....	25
4.7 <mpvpTrans:TransitionFilterProperties>.....	26
4.7.1 Property: mpvpTrans:Dur .....	26
4.7.2 Property: mpvpTrans:Subtype .....	27
4.7.3 Property: mpvpTrans:Type.....	27
4.7.4 Recommended Transitions .....	28
Chapter 5: MPV Presentation Module Practices .....	29
5.1 Best Practices for Presenting a Manifest.....	29
5.2 Best Practices for Watching .....	29

- 5.3 Best Practices for Browsing ..... 30
- 5.4 Best Practices for Supported Formats..... 31
- 5.5 Examples ..... 31
  - 5.5.1 Startup List of Albums with Background Image ..... 31
  - 5.5.2 Representing a Title Image ..... 33
  - 5.5.3 Album Renditions of a Video Slideshow and Printed Content ..... 33
  - 5.5.4 Building Up a StillSequenceWithAudio Type..... 33
- Appendix I: Transition Types Reference ..... 34
- Appendix II: Typographic Conventions ..... 37
- Appendix III: References ..... 38

# Chapter 1: Introduction

## 1.1 Executive Summary

MultiPhoto/Video (MPV) is an open specification that makes easier the representation, exchange, 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 [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, such 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 supports use of established specifications and standards.

The development and promotion of MultiPhoto/Video is sponsored by the Optical Storage Technology Association (OSTA). The specification development and promotion process is open to all members; all organizations and individuals are welcomed as members. The association includes over 50 member companies from all over the world that produce products that collectively represent a majority marketshare in mainstream recordable optical storage categories.

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

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”, 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.

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 may be augmented by information from various profiles. For example, the Presentation profile provides information that may be used by player applications and devices to provide an attractive playback 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 without royalty from the Optical Storage Technology Association (OSTA). OSTA is an industry association promoting the use and interoperability of recordable CD and DVD discs in computer and consumer electronics devices.

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 provides for the basic definition of collections of photo-video content. A Presentation Profile extends the specification for an enhanced presentation experience of interactive browsing and slideshow playback provided by DVD players and other devices and media player software. Additional profiles will be developed subsequently.

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 requires the device or application to reveal 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 [SMIL20] and I3A's DIG35 [DIG35-2001]. MPV is compatible with and supports the DCF [DCF-1999] and Exif [Exif2002] specifications that are widely used in digital cameras.

### MultiPhoto/Video Provides ...

#### Fast and friendly user experience:

- Start fast, play slideshow, browse interactively
- Interact with albums and photo-video “items”, not files.
- Robust against renaming and reorganization of files

#### One format that can work anywhere:

- Any storage media, any device, any software
- “Adds on” to existing technology without conflict.
- Fully extensible

#### Immediate Value:

- Playback on Microsoft Win XP and IE 5.5+ browsers or choice of other players.

#### Future Value:

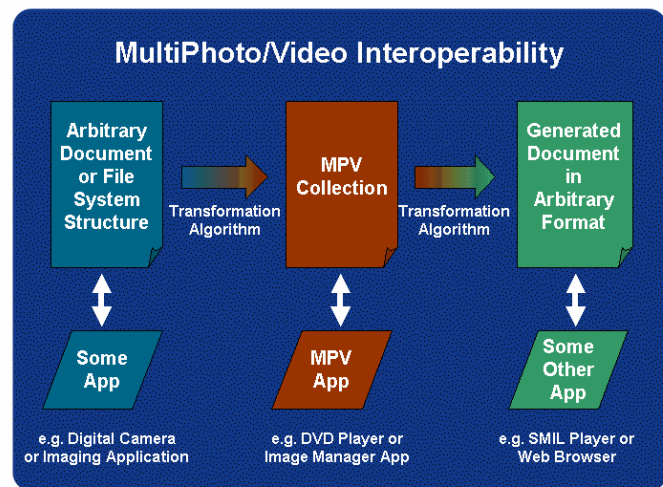
- Playback on DVD players and other devices supporting MPV [target: Christmas 2002]

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 as "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 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.

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



# Chapter 2: Key Concepts of the MPV and Related Specifications

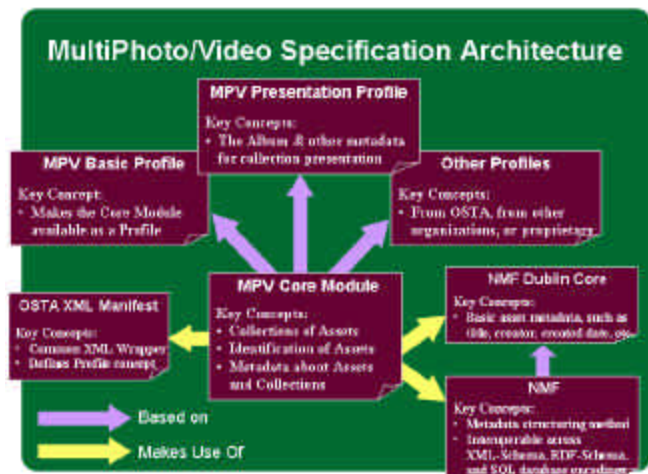
## 2.1 MPV Specification Architecture

MPV is not just one specification, it is a family of related specifications. This architecture results from several principle objectives:

- MPV should be highly extensible, allowing anyone to create proprietary or open extensions to MPV without modifying the MPV specification itself.
- Adding extensions should not damage interoperability of the basic collection information.
- Specifications that are fundamentally separable concepts should be separated. This allows each specification to be used and evolve independently of each other.
- MPV should not define alternate representations where mainstream representations exist.

These principles result in the following characteristics of the MPV and related specifications.

- The **MPV Core** is the essence of the MPV specification. However, it cannot be used by itself; it must always be incorporated into a Profile, which is the basic unit of extension in MPV.
- The **MPV Basic and Presentation Profiles** are extensions that utilize the MPV Core. **Other Profiles** are extensions organized in exactly the same way.
- MPV makes use of the **OSTA XML Manifest**, which defines the Profile concept.
- MPV makes use of the **NMF Specification** for structured representation of arbitrary metadata. NMF is a wholly separate concept.
- MPV recommends use of the NMF-encoding of **Dublin Core**, a separate and widely adopted specification for representing basic metadata about assets of all kinds.



## 2.2 Profiles and Modules, Schema and Practices

The MultiPhoto/Video specifications contain the following kinds of content.

**Schema** define the structure of MPV content, providing 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 additional content and are the principle 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.

**Referenced Specifications** are other specifications used by the MPV specifications. These specifications may be from OSTA or other organizations.

## 2.3 MPV Core and the MPV Basic and Presentation Profiles

Profiles represent the basic unit of extension within MPV. Profiles define schema and practices that are available for addressing a given task. Typically, profiles define one or more top-level elements in a manifest plus various additional metadata.

The MPV Core, Basic, and Presentation Profiles were all developed at one time, and the Core documentation makes reference to the Basic and Presentation Profiles. However, nothing about the Basic and Presentation Profile specifications and implementation is treated specially in the Core. In other words, the Basic and Presentation Profiles play by the same rules that new, as yet undefined additional profiles must play by.

### 2.3.1 MPV Core

An overview of the MPV Core is described in the MPV Core specification [MPV-Core]. It provides the three core concepts of Collections, Metadata, and Identification. No profile is considered an MPV-related Profile unless it makes use of the MPV Core, either directly or indirectly.

### 2.3.2 MPV Basic Profile

This specification defines The MPV Basic Profile which simply makes available the MPV Core as an MPV Profile. This is necessary since Profiles are the only unit of incorporation into an OSTA XML Manifest. The MPV Basic Profile provides two top-level elements in a manifest: AssetList and MarkList.

Primary tasks that users of the MPV Basic Profile can accomplish include definition of collections of assets by reference and grouping of those assets into distinguished sets using MarkLists. These basic tasks are the essence of what MPV provides.

## **ASSETLIST**

The AssetList is the basic unit of collection representation in MPV. Assets themselves may only be defined in an AssetList. Only one AssetList is allowed in a manifest.

## **MARKEDASSETS**

The MarkedAssets element may contain MarkLists that make reference to assets. Standard marklist types like “primary” and “selected” provide for interchange of lists of distinguished assets. Multiple MarkLists may be present in the MarkedAssets element.

### **2.3.3 MPV Presentation Profile**

Primary tasks for the MPV Presentation Profile are to provide albums that allow the user apply presentation information to a collection of assets. The primary usages are to play a slideshow, interactively browse the primary assets, or selectively print the album’s contents.

The MPV Presentation Profile provides very basic presentation information that emphasizes use by a devices and applications with a broad range of presentation capabilities and significant amounts of application-level control over presentation behaviour. Additional profiles may define much richer or more tightly scoped presentation behaviours.

## **ALBUM, ALBUMREF**

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 contain references to other albums using the AlbumRef construct. These other albums can either be in the same of different manifest as the reference. Multiple albums can be grouped together in one file or isolated in separate files. Albums link to other albums using the AlbumRef construct which uses the standard MPV referencing mechanisms to reference either 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.

## **PRESENTATION CONTROL**

The overall approach for representing presentation information is compatible with SMIL, a powerful XML format for representing presentations from the World Wide Web Consortium (W3C). MPV Presentation Profile provides a very constrained set of properties compatible with 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 the Presentation Profile is not extensive however, many other implementations can be contemplated. For example, compelling playback of MPV documents in modern web browsers is readily accomplished.

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

## 2.4 XML Usage

### XML LEVERAGE

MPV content is well-formed XML. This allows the MPV document to be processed using standard XML processing tools and environments. For example, when opened in the Microsoft Internet Explorer 5.5 and above web browser, a 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. MPV can also be easily utilized within other XML specifications.

### NAMESPACES

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

MPV requires the use of namespaces. By convention, all elements and attributes in MPV are used with their prefixes in all XML encodings. MPV does not support namespace-unaware processing. Most modern XML tools support namespace-aware processing.

## 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 Schema 1.0
- MPV Core Practices 1.0
- MPV Presentation Schema 1.0
- MPV Presentation 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
Presentation Profile	<a href="http://ns.osta.org/mpv/presentation/1.0/">http://ns.osta.org/mpv/presentation/1.0/</a>	profile/basic/basic.xsd	mpv:

# Chapter 4: MPV Presentation Schemas

## 4.1 Introduction

The MPV Presentation schemas 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
- NMF Metadata:
  - Media Presentation Control properties – presentation-related information, such as duration
  - Transition Filter – transition effects between assets

## 4.2 Schema Information

The MPV presentation module uses the following schemas:

Schema group	Namespace Identifier	Conventional Namespace Prefix
Presentation	<a href="http://ns.osta.org/mpv/presentation/1.0/">http://ns.osta.org/mpv/presentation/1.0/</a>	mpvp:
Control	<a href="http://ns.osta.org/mpv/presentation/1.0/Control/">http://ns.osta.org/mpv/presentation/1.0/Control/</a>	mpvpCtrl:
Sequence	<a href="http://ns.osta.org/mpv/presentation/1.0/Sequence/">http://ns.osta.org/mpv/presentation/1.0/Sequence/</a>	mpvpSeq:
TransitionFilter	<a href="http://ns.osta.org/mpv/presentation/1.0/TransitionFilter/">http://ns.osta.org/mpv/presentation/1.0/TransitionFilter/</a>	mpvpTrans:

## 4.3 <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.

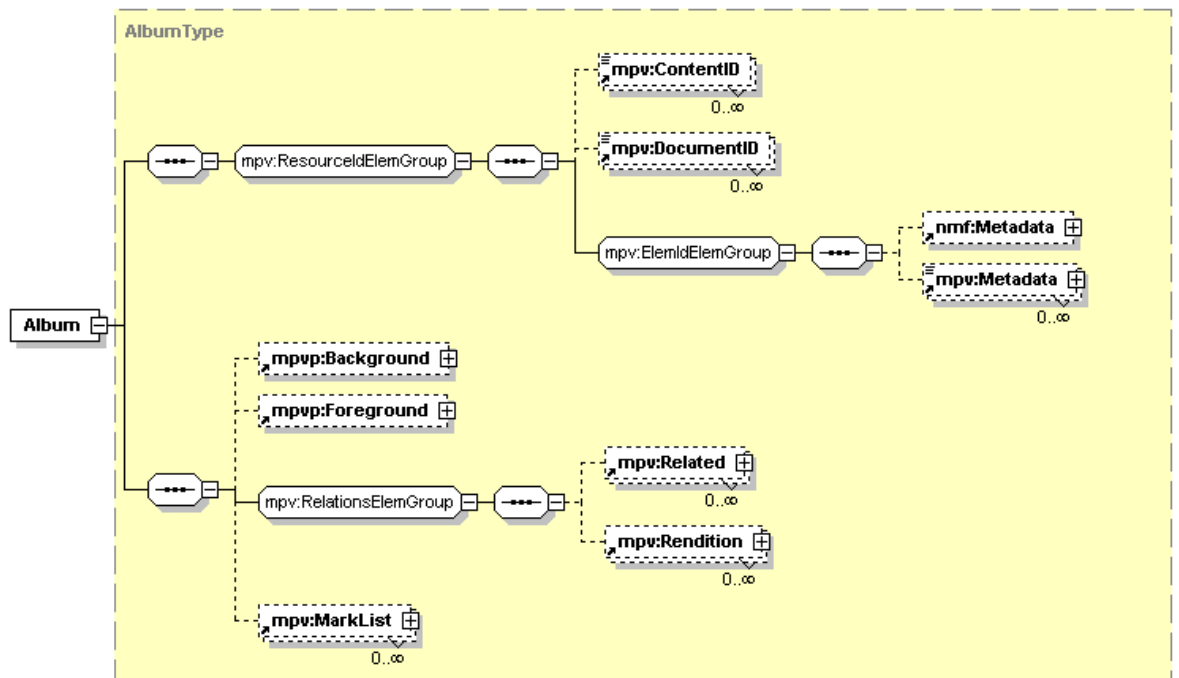
Arbitrary data can be attached to a collection, but it carries no explicit semantics with respect to MPV collection processing other than the data is associated with the collection. The Data element can make reference to data files associated with the Album but which are not media assets or are not Foreground 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.

### 4.3.1.1 element element Album

diagram



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

type [AlbumType](#)

children [mpv:ContentID](#) [mpv:DocumentID](#) [Metadata](#) [mpv:Metadata](#) [Background](#) [Foreground](#) [mpv:Related](#) [mpv:Rendition](#) [mpv:MarkList](#)

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:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
    
```

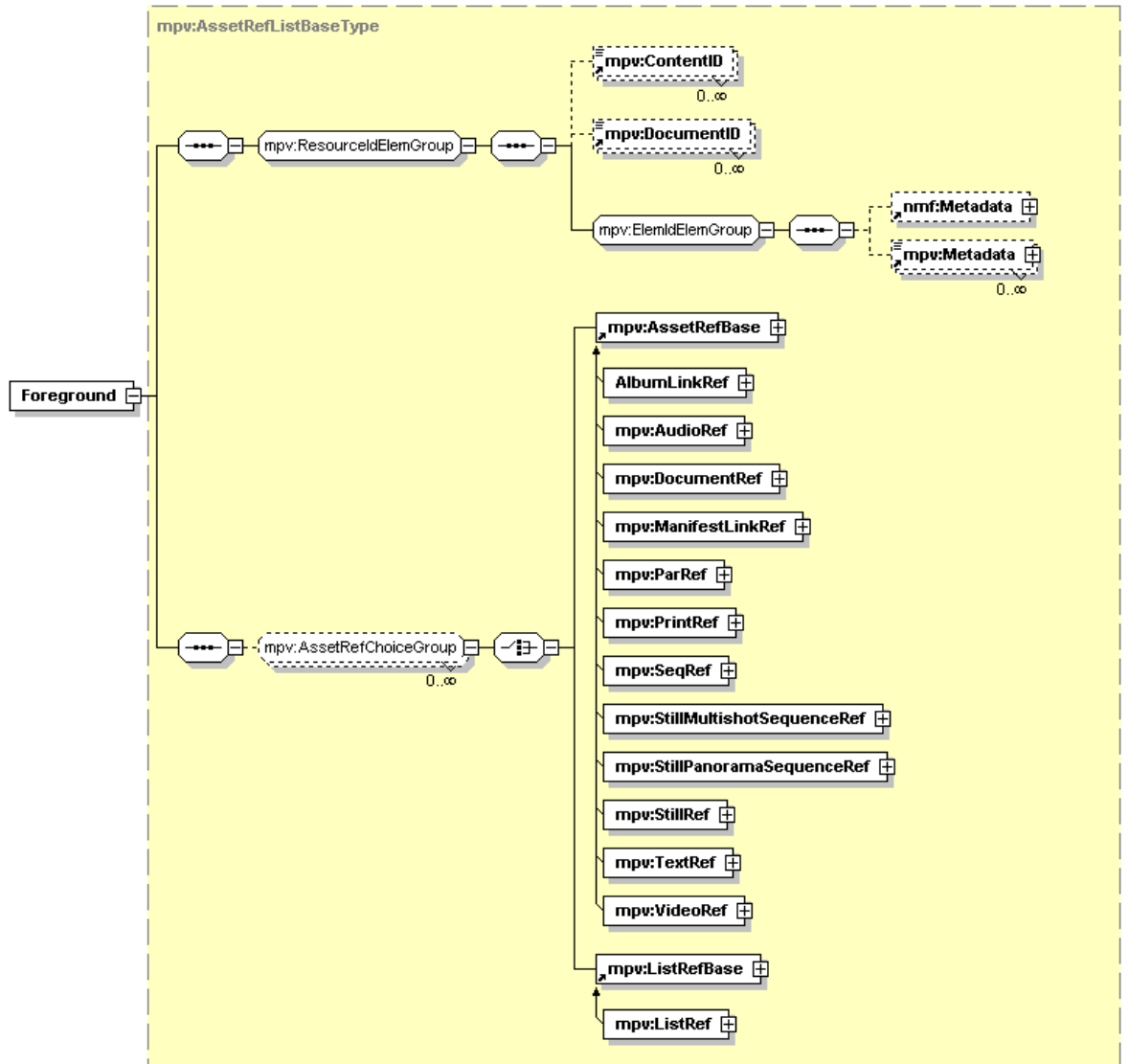
```
<xs:group ref="mpv:RelationsElemGroup"/>
<xs:element ref="mpv:MarkList" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
```

## 4.4 <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 **Foreground**  
diagram



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

type **mpv:AssetRefListBaseType**

children **mpv:ContentID mpv:DocumentID Metadata mpv:Metadata mpv:AssetRefBase mpv:ListRefBase**

used by complexType **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		
	defaultManifestLinkIDRef	xs:IDREF	optional		

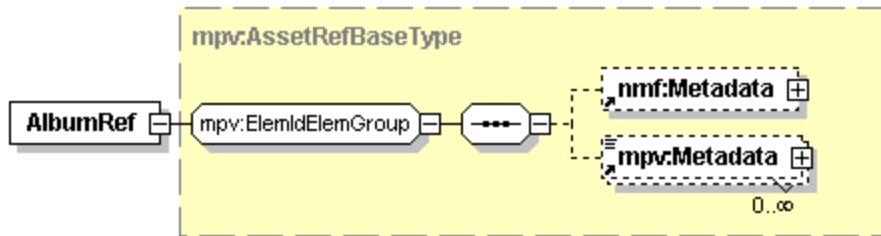
The Background element diagram is identical.

## 4.5 <mpvp:AlbumRef>

An AlbumRef element references an Albumasset in either the same or a different manifest.

### element AlbumRef

diagram



namespace <http://ns.osta.org/mpvp/presentation/1.0/>

type [mpv:AssetRefBaseType](#)

children [Metadata](#) [mpv:Metadata](#)

attributes	Name	Type	Use	Default	Fixed
	manifestLinkIDRef	xs:IDREF	optional		
	listIDRef	xs:IDREF	optional		
	mpv:id	xs:ID			
	idRef	xs:Name	required		

source `<xs:element name="AlbumRef" type="mpv:AssetRefBaseType" substitutionGroup="mpv:AssetRefBase"/>`

## 4.6 <mpvpCtrl:ControlProperties> Media Asset Presentation Control

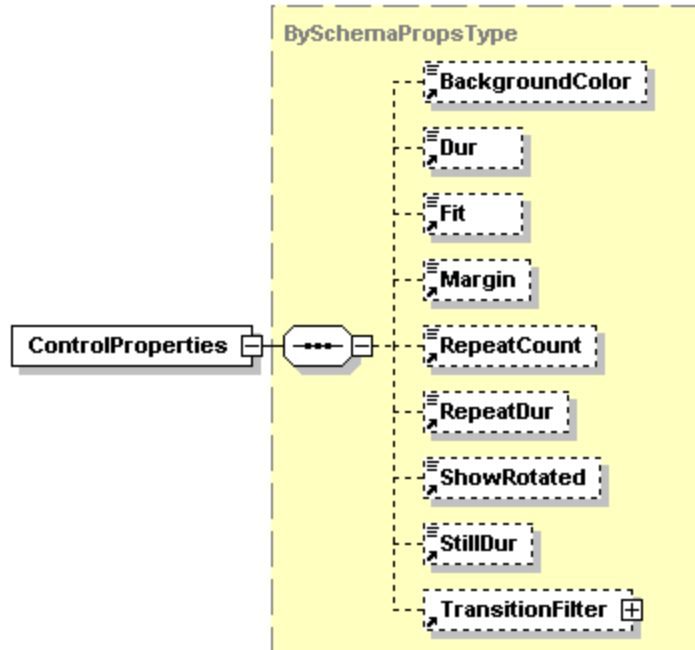
The Presentation Profile defines a schema for presentation properties. This schema can be used on all media assets by specifying the root element of the mpvp schema as the only child of the vtmp:Metadata element.

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 implemented 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 <mpvp:Foreground> element itself will apply to all its children.

## element **ControlProperties**

diagram



namespace <http://ns.osta.org/mpvp/presentation/1.0/Control/>

type [BySchemaPropsType](#)

children [BackgroundColor](#) [Dur](#) [Fit](#) [Margin](#) [RepeatCount](#) [RepeatDur](#) [ShowRotated](#) [StillDur](#) [TransitionFilter](#)

source `<xs:element name="ControlProperties" type="BySchemaPropsType" substitutionGroup="nmf:BySchemaPropsBase"/>`

```

source <xs:complexType name="BySchemaPropsType">
  <xs:complexContent>
    <xs:extension base="nmf:BySchemaPropsType">
      <xs:sequence>
        <xs:element ref="BackgroundColor" minOccurs="0"/>
        <xs:element ref="Dur" minOccurs="0"/>
        <xs:element ref="Fit" minOccurs="0"/>
        <xs:element ref="Margin" minOccurs="0"/>
        <xs:element ref="RepeatCount" minOccurs="0"/>
        <xs:element ref="RepeatDur" minOccurs="0"/>
        <xs:element ref="ShowRotated" minOccurs="0"/>
        <xs:element ref="StillDur" minOccurs="0"/>
        <xs:element ref="TransitionFilter" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

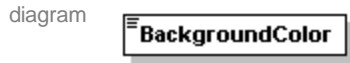
```

### 4.6.1 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,

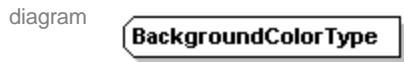
the implementation dependent window background will be shown. The vocabulary for BackgroundColor is defined by CSS2 system colors [CSS2].

**element BackgroundColor**



namespace <http://ns.osta.org/mpv/presentation/1.0/Control/>  
 type [BackgroundColorType](#)  
 used by complexType [BySchemaPropsType](#)  
 source `<xs:element name="BackgroundColor" type="BackgroundColorType"/>`

**complexType BackgroundColorType**



namespace <http://ns.osta.org/mpv/presentation/1.0/Control/>  
 type extension of `xs:string`  
 used by element [BackgroundColor](#)  
 source `<xs:complexType name="BackgroundColorType">  
 <xs:simpleContent>  
 <xs:extension base="xs:string"/>  
 </xs:simpleContent>  
 </xs:complexType>`

## 4.6.2 Properties: mpvpCtrl:Dur

Specifies the simple duration.

**element Dur**



namespace <http://ns.osta.org/mpv/presentation/1.0/Control/>  
 type [DurType](#)  
 used by complexType [BySchemaPropsType](#)  
 source `<xs:element name="Dur" type="DurType"/>`

**simpleType DurationType**

namespace	<a href="http://ns.osta.org/mpv/presentation/1.0/Control/">http://ns.osta.org/mpv/presentation/1.0/Control</a>
type	<code>xs:string</code>
used by	element <a href="#">Properties/Dur</a>
source	<code>&lt;xs:simpleType name="DurationType"&gt;          &lt;xs:restriction base="xs:string"/&gt;          &lt;/xs:simpleType&gt;</code>

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"**

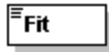
Specifies the simple duration as indefinite.

**4.6.3 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 Fit**

diagram



namespace <http://ns.osta.org/mpvp/presentation/1.0/Control/>

type [FitType](#)

used by complexType [BySchemaPropsType](#)

source `<xs:element name="Fit" type="FitType"/>`

**simpleType FitType**

namespace	<a href="http://ns.osta.org/mpvp/presentation/1.0/Control/">http://ns.osta.org/mpvp/presentation/1.0/Control/</a>
type	union of ( <a href="#">FitBaseType</a> , <a href="#">xs:anyURI</a> )
used by	element <b>Properties/Fit</b>
source	<code>&lt;xs:simpleType name="FitType"&gt; &lt;xs:union memberTypes="mpvpCtrl:FitBaseType xs:anyURI"/&gt; &lt;/xs:simpleType&gt;</code>

**simpleType FitBaseType**

namespace	<a href="http://ns.osta.org/mpvp/presentation/1.0/Control/">http://ns.osta.org/mpvp/presentation/1.0/Control/</a>
type	restriction of <a href="#">xs:string</a>
used by	simpleType <b>FitType</b>
facets	enumeration <b>hidden</b> enumeration <b>fill</b> enumeration <b>meet</b> enumeration <b>scroll</b> enumeration <b>slide</b>
source	<code>&lt;xs:simpleType name="FitBaseType"&gt; &lt;xs:restriction base="xs:string"&gt; &lt;xs:enumeration value="hidden"/&gt; &lt;xs:enumeration value="fill"/&gt;</code>

	<pre>&lt;xs:enumeration value="meet"/&gt; &lt;xs:enumeration value="scroll"/&gt; &lt;xs:enumeration value="slide"/&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt;</pre>
--	---

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.

## **4.6.4 Properties: mpvpCtrl:Margin**

Specifies the percentage of the percentage of the height and width of the region that should be allocated as a margin area around the media asset element. This value is specified as a floating point number between 0 and 1.

### element **Margin**

diagram



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

type [MarginType](#)

used by complexType [BySchemaPropsType](#)

source `<xs:element name="Margin" type="MarginType"/>`

complexType **MarginType**



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

type extension of **xs:float**

used by element [Margin](#)

```
<xs:complexType name="MarginType">
  <xs:simpleContent>
    <xs:extension base="xs:float"/>
  </xs:simpleContent>
</xs:complexType>
```

### 4.6.5 Properties: mpvpCtrl:RepeatCount

Specifies the number of iterations of the simple duration.

element **Properties/RepeatCount**

diagram	
namespace	<a href="http://ns.osta.org/mpv/presentation/1.0/Control/">http://ns.osta.org/mpv/presentation/1.0/Control/</a>
type	<b>xs:string</b>
source	<code>&lt;xs:element name="RepeatCount" type="xs:string" minOccurs="0"/&gt;</code>

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).

### 4.6.6 Properties: mpvpCtrl:RepeatDur

Specifies the total duration for repeat.

element **Properties /RepeatDur**

diagram	
namespace	<a href="http://ns.osta.org/mpv/presentation/1.0/Control/">http://ns.osta.org/mpv/presentation/1.0/Control/</a>
type	<b>xs:string</b>
source	<code>&lt;xs:element name="RepeatDur" type="xs:string" minOccurs="0"/&gt;</code>

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).

**4.6.7 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.

**4.6.8 Properties: mpvpCtrl:StillDur**

Specifies, in seconds, the length of time that Stills should be displayed during a sequential presentation of assets.

element **StillDur**

diagram



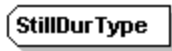
namespace http://ns.osta.org/mpv/presentation/1.0/Control/

type [StillDurType](#)used by complexType [BySchemaPropsType](#)

source &lt;xs:element name="StillDur" type="StillDurType"/&gt;

complexType **StillDurType**

diagram



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

type extension of xs:float

used by element [StillDur](#)

```
source <xs:complexType name="StillDurType">
  <xs:simpleContent>
    <xs:extension base="xs:float"/>
  </xs:simpleContent>
</xs:complexType>
```

**4.6.9 Property: mpvpCtrl:TransitionFilter**

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



element **TransitionFilter**

diagram

namespace <http://ns.osta.org/mpvp/presentation/1.0/Control/>type [TransitionFilterType](#)children [TransitionFilterProperties](#)used by complexType [BySchemaPropsType](#)source `<xs:element name="TransitionFilter" type="TransitionFilterType"/>`

```

source <xs:complexType name="TransitionFilterType">
  <xs:complexContent>
    <xs:extension base="nmf:CompositePropType">
      <xs:sequence>
        <xs:element ref="mpvpTrans:TransitionFilterProperties"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

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

#### 4.6.10 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.

```

<file:Manifest
  xmlns:file="http://ns.osta.org/manifest/1.0/"
  xmlns:mpv="http://ns.osta.org/mpv/1.0/"
  xmlns:nmf="http://ns.osta.org/nmf/1.0/"
  xmlns:mpvpCtrl="http://ns.osta.org/mpvp/presentation/1.0/Control/"
  xmlns:mpvpTrans="http://ns.osta.org/mpvp/presentation/1.0/TransitionFilter/"
>
  . . .
  <mpv:Still>
    . . .
    <nmf:Metadata>
      <ControlProperties xmlns="http://ns.osta.org/mpvp/presentation/1.0/Control/">
        <BackgroundColor>Blue</BackgroundColor>
        <Fit>meet</mpvpCtrl:Fit>
        <TransitionFilter>
          <TransitionFilterProperties
            xmlns="http://ns.osta.org/mpvp/presentation/1.0/mpvpTrans">
            <Type>barWipe</Type>
          </TransitionFilterProperties>
        </TransitionFilter>
      </ControlProperties>
    </nmf:Metadata>

```

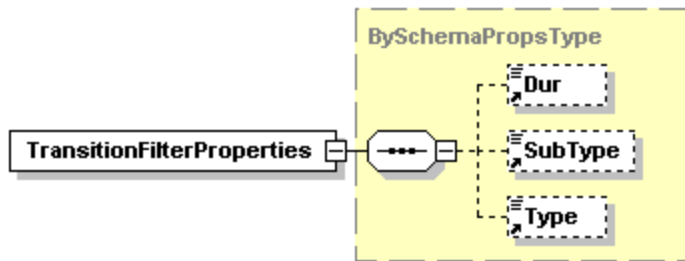
```
</mpv:Still>
</file:Manifest>
```

## 4.7 <mpvTrans:TransitionFilterProperties>

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 TransitionFilterProperties

diagram



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

type [BySchemaPropsType](#)

children [Dur](#) [SubType](#) [Type](#)

used by complexType [TransitionFilterType](#)

source `<xs:element name="TransitionFilterProperties" type="BySchemaPropsType" substitutionGroup="nmf:BySchemaPropsBase"/>`

```

source <xs:complexType name="BySchemaPropsType">
  <xs:complexContent>
    <xs:extension base="nmf:BySchemaPropsType">
      <xs:sequence>
        <xs:element ref="Dur" minOccurs="0"/>
        <xs:element ref="SubType" minOccurs="0"/>
        <xs:element ref="Type" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

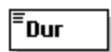
```

### 4.7.1 Property: mpvTrans:Dur

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

#### element Dur

diagram



namespace <http://ns.osta.org/mpv/presentation/1.0/TransitionFilter/>  
 type [DurType](#)  
 used by complexType [BySchemaPropsType](#)  
 source `<xs:element name="Dur" type="DurType"/>`  
 source `<xs:complexType name="DurType">  
 <xs:simpleContent>  
 <xs:extension base="xs:float"/>  
 </xs:simpleContent>  
 </xs:complexType>`

### 4.7.2 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 **SubType**



namespace <http://ns.osta.org/mpv/presentation/1.0/TransitionFilter/>  
 type [SubTypeType](#)  
 used by complexType [BySchemaPropsType](#)  
 source `<xs:element name="SubType" type="SubTypeType"/>`  
 source `<xs:complexType name="SubTypeType">  
 <xs:simpleContent>  
 <xs:extension base="xs:string"/>  
 </xs:simpleContent>  
 </xs:complexType>`

### 4.7.3 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 **Type**



namespace <http://ns.osta.org/mpv/presentation/1.0/TransitionFilter/>  
 type [TypeType](#)  
 used by complexType [BySchemaPropsType](#)  
 source `<xs:element name="Type" type="TypeType"/>`  
 source `<xs:complexType name="TypeType">  
 <xs:simpleContent>  
 <xs:extension base="xs:string"/>  
 </xs:simpleContent>  
 </xs:complexType>`

#### **4.7.4 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.

<b>Transition type</b>	<b>Default Transition subtype</b>
barWipe	leftToRight
irisWipe	rectangle
clockWipe	clockwiseTwelve
snakeWipe	topLeftHorizontal

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

# 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

An OSTA XML manifest [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 sequence 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 Album, the recommended behaviour for the watch application is to skip silently over the Album.

## 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.

## ALBUMREF

When the Album contains references to Albumassets, 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

This is a complete example of a startup list of 3 albums in the foreground of the index album. The background has a single reference to a Still asset that is described in the assetlist of the same manifest that contains the first album.

```
<?xml version="1.0" encoding="UTF-8"?>
<file:Manifest
  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:nmf="http://ns.osta.org/nmf/1.0/" >
<nmf:Metadata>
  <Properties 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>
    </ProfileBag>
  </Properties>
</nmf:Metadata>

<mpvp:Album>
  <mpvp:Background>
    <mpvp:StillRef mpv:manifestLinkIDRef="000100" mpv:idRef="002" />
  </mpvp:Background>
  <mpvp:Foreground>
    <mpv:AlbumRef mpv:manifestLinkIDRef="000100" mpv:idRef="001" />
    <mpv:AlbumRef mpv:manifestLinkIDRef="000200" mpv:idRef="001" />
    <mpv:AlbumRef mpv:manifestLinkIDRef="000300" mpv:idRef="001" />
  </mpvp:Foreground>
</mpvp:Album>

<mpv:AssetList>

  <!-- AlbumLink -->
  <mpv:ManifestLink mpv:id="000100">
    <mpv:LastURL mpv:filesystem="Joliet">Collection 1 - stills asset
list/album.mpv</mpv:LastURL>
    <mpv:LastURL mpv:filesystem="NTFS">Collection 1 - stills asset
list/album.mpv</mpv:LastURL>
    <mpv:ContentID>urn:osta-
org:mpv:dsig:all:md5:EF886AEFA3B340da971BAF09B17DBC122</mpv:ContentID>
    <nmf:Metadata>
      <Properties xmlns="http://purl.org/dc/1.0">
        <title>Collection 1 - stills asset list</title>
      </Properties>
    </nmf:Metadata>
  </mpv:ManifestLink>

  <!-- ManifestLink -->
  <mpv:ManifestLink mpv:id="000200">
    <mpv:LastURL mpv:filesystem="Joliet">Collection 2 - stills
album/album.mpv</mpv:LastURL>
    <mpv:LastURL mpv:filesystem="NTFS">Collection 2 - stills
album/album.mpv</mpv:LastURL>
    <mpv:ContentID>urn:osta-
org:mpv:dsig:all:md5:EF886AEFA3B340da971BAF09B17DBC122</mpv:ContentID>
    <nmf:Metadata>
      <Properties xmlns="http://purl.org/dc/1.0">
        <title>Collection 2 - stills album</title>
      </Properties>
    </nmf:Metadata>
  </mpv:ManifestLink>

  <!-- ManifestLink -->
  <mpv:ManifestLink mpv:id="000300">

```



```
<mpv:LastURL mpv:filesystem="Joliet">Collection 4 - full multimedia
album</mpv:LastURL>
<mpv:LastURL mpv:filesystem="NTFS">Collection 4 - full multimedia album</mpv:LastURL>
<mpv:ContentID>urn:osta-
org:mpv:dsig:all:md5:EF886AEFA3B340da971BAF09B17DBC122</mpv:ContentID>
<nmf:Metadata>
  <Properties xmlns="http://purl.org/dc/1.0">
    <title>Collection 4 - full multimedia album</title>
  </Properties>
</nmf:Metadata>
</mpv:ManifestLink>

</mpv:AssetList>

</file:Manifest>
```

### **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

---

Examples of MPV metadata structures are in Courier font.

```
<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/>

**[DC-NMF]**

"Dublin Core Normalized Metadata Format Profile Specification 1.0"; OSTA, 19 June 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, 19 June 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>

**[MPV-Basic]**

"MultiPhoto/Video – Basic Profile Specification", OSTA, 19<sup>th</sup> June, 2002,  
Available at <http://www.osta.org/mpv/>

**[MPV-CORE]**

"MultiPhoto/Video Core Specification 1.0"; OSTA, 19 June 2002,.  
Available at <http://www.osta.org/mpv/>

**[NMF]**

"Normalized Metadata Format Specification 1.0"; OSTA, 19 June 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>

**[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]**

"VXMP – 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/>