



*Digital Music, Photo, and Video Collections*



# MPV Music Profile Specification

**Revision 0.70  
Working Draft**

**21 January 2003**

**© 2002-2003 Optical Storage Technology Association**

## **IMPORTANT NOTICE**

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

**NOT FOR DISBTRIBUTION ON A PUBLIC WEBSITE**

This document is available at <http://www.osta.org/mpv/mpvmbrr/specs/MPVMusicProf-Spec-0.70WD.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>MPV 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, MPV Specification MPV Initiative Lead</p> <p>Tel: +1 541-715-8658 Email: Pieter_van_Zee@hp.com</p> <p>Felix Nemirovsky Chairman, MultiRead Subcommittee</p> <p>Tel: +1 415 643 0944 Email: felixn@pacbell.net</p>
---	---

### ABSTRACT

The Music Profile specification defines metadata and practices for processing and playback of collections of digital music collections stored on an optical disc and other storage media such as memory cards and computer harddrives or exchanged via internet protocols.

### COPYRIGHT NOTICE

Copyright 2002-2003 Optical Storage Technology Association, Inc.

## **LICENSING IMPORTANT NOTICES**

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

# Contents

---

Contents .....	4
Chapter 1: Introduction .....	6
1.1 Executive Summary .....	6
1.2 Terms of Use .....	7
Chapter 2: MPV Music Profile 1.0.....	8
2.1 MPV Music Profile Introduction.....	8
2.2 Formalities For Use of the MPV Music Profile .....	9
Chapter 3: MPV Music Schema Introduction.....	11
3.1 Introduction.....	11
3.2 Examples .....	11
3.2.1 Namespaces and Profiles .....	11
3.2.2 Simple Example .....	12
3.2.3 Rich Example .....	13
3.3 Use of Existing MPV Specifications .....	16
3.4 MPV Music Profile Metadata Introduction .....	16
3.5 Use of Dublin Core Metadata .....	18
3.6 Music Media Types .....	19
Chapter 4: MPV Music Profile – Using MPV Playlists .....	20
4.1 MPV Music Playlists.....	20
4.1.1 One Playlist per MPV File.....	20
4.1.2 Metadata Usage .....	21
4.1.3 Background Usage .....	21
4.1.4 Foreground Usage.....	21
4.1.5 Related and Rendition Assets .....	22
4.2 Linked Playlists .....	22
4.3 Default Playlists.....	24
4.4 Dynamic Playlists .....	25
4.4.1 Specifying Dynamic Playlists .....	25
4.4.2 Playlist Query Language.....	25
4.5 Best Practices for Generating Playlists .....	25
4.5.1 Top-Level Playlist.....	26
4.5.2 “All Music” Playlist.....	26
4.5.3 Local mpv:AssetList .....	26
4.5.4 Centralized mpv:AssetList .....	26
Chapter 5: MPV Music Schemas In Detail .....	27
5.1 Multiple Renditions of a Music Asset.....	27
5.2 Artwork for an Asset .....	29
5.3 Asset-related Content .....	30
5.4 Music-specific Metadata .....	30

5.5 Album/Playlist-level <mpvm:MusicProperties> Music Metadata .....35

5.6 MPV Music Profile Example.....35

Chapter 6: MPV Music Profile Mapping To Other Music Metadata Formats .....38

6.1 ID3 and OSTA MPV Music Profile .....38

6.2 WinAMP M3U and OSTA MPV Music Profile .....39

6.3 OSTA MultiAudio and OSTA MPV Music Profile.....39

Appendix I: References .....41

# Chapter 1: Introduction

## 1.1 Executive Summary

MPV (Music/Photo/Video) is an open specification that makes easier the representation, exchange, processing and playback of collections of digital media content, including music, still images, stills with audio, still sequences, video clips, and audio clips.

Applications and devices and users that use MPV benefit even when they only interact with music and audio in basic ways; such as personal music collections that can be burned on CDs by many software applications.

MPV uses a simple text-based format that is easily understood and also easy to produce and consume programmatically in firmware or computer software. MPV 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 MPV 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.

MPV 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 are being contributed by interested parties.

The specification is being developed in phases and results in "profiles". Each profile in MPV 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
- **Music Profile:** key tasks: listening to a music collection and interactively browsing content collections
- **Internet Profile:** key task: interacting with and sending collections of photo-video content over the web and email
- **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. The Presentation Profile is also used by the Music Profile to as a music playlist.

MPV technology has three central components: Collections, Metadata, and Identification. Each of these make reference in various ways to data files containing the music, photo, or 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 Terms of Use

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

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

MPV 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. In addition, some sample open-source code implementations of key steps in processing MPV content may be contributed by interested parties.

## Chapter 2: MPV Music Profile 1.0

The MPV Music Profile allows users, via applications and devices, to create and playback collections of music organized into albums and playlists. The MPV Music Profile extends the existing MPV Core specification and Basic and Presentation Profiles by augmenting this framework with additional metadata and practices specific to music.

A user may organize their music content into collections and burn it on a recordable CD or DVD. When the music collections are represented on the disc using MPV files that implement the MPV Music Profile, then a playback application or device can quickly start playback as soon as the disc is inserted and allow the user to easily navigate and playback the music, equally well and quickly regardless of whether the disc has 15, 150, or 1500 songs on it. Of course, in addition to basic music playback, an application or device can show basic music information like title, artist and genre – this may be retrieved from the music files themselves or from the MPV collection. Additional content may also be part of the collection and is available to be shown by the playback application or device, such as artwork related to the music, lyrics, and even music videos of the songs.

The Music Profile provides a basic set of metadata which represents data and conventions used by software applications that create and play compressed audio music on PCs or consumer electronics devices and music publishers of music CDs. The MultiAudio specification, already developed by OSTA [**MultiAudio**], is an earlier generation of technology specific to audio. MPV integrates music, photos, and video and can be used to create and exchange multimedia playlists and collections.

### 2.1 MPV Music Profile Introduction

The MPV Music Profile 1.0 supports the following key tasks: defining collections of music, organizing music into albums and playlists, listening sequentially or shuffled to an album / playlist, and interactively browsing single or multiple album / playlists of music.

The music metadata that may be represented using the MPV Music Profile includes the following:

**Music Asset (“Song”, “Track”):** Filename, Title, Principal artist, Album title, Genre, Playing time, Date recorded, Order on original album, Artwork, Music videos, Performed by, Music by, Lyrics by, Arranged by, More info URL, Average encoded bitrate, Lyrics, Rights, Identifier, Description, Format,

The MPV Music Profile can also organize music content in useful and novel ways. For example, a music asset may have multiple representations, such as multiple bitrate encodings, multiple format encodings (which may enhance compatibility with devices), and multiple representations, such as audio-only, video-with-audio, and song artwork.

**Album (“Playlist”) of Music:** Title, Principal Artist, Description, Identifier, Artwork, Music Entries



The capabilities of the MPV Music Profile allows discs to be produced that have variable user experiences depending on the type of device used to play them. For example, a low-cost CD player could just play MP3 music and display information on a 4 line LCD display. A capable DVD player could play music videos and display music information on a multi-line graphical display along with artwork and lyrics.

MPV also allows music to be organized into hierarchical playlists, allowing users to navigate among playlists that may be both pre-generated or created on-the-fly by the playback application.

## 2.2 Formalities For Use of the MPV Music Profile

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

### PROFILE COMPONENTS

The MPV Music Profile 1.0 makes use of two other specifications:

- MPV Core Specification 1.0
- MPV Presentation Profile Specification 1.0

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

### COMPATIBILITY

The MPV Music Profile 1.0 is an extension of the MPV Core Specification 1.0 and is fully compatible with the MPV framework it establishes. Thus MPV files that implement the MPV Music Profile should be usable in basic ways by MPV-aware applications and devices not focused on music playback. This means, for example, that a MPV playback application or device can read and playback MPV music collections even if it doesn't understand the MPV Music Profile; however, the music-specific information will be ignored and the playback experience will be less full-featured than in a MPV Music Profile player.

### SCHEMA NAMESPACE

To use the MPV Music Profile, this information must be present in the namespace declarations in the MPV file:

Schema	Namespace Identifier	Schema Location	Conventional Namespace Prefix
Music Profile	<a href="http://ns.osta.org/mpv/music/1.0/">http://ns.osta.org/mpv/music/1.0/</a>	lax/profiles/music/profile.xsd	mpvm:

The schema location may be specified optionally. Multiple schema variations may exist depending on the degree of validation desired by the developer. Typical variations include "lax", "strict", and "fixed". These schema will all implement the grammar of the MPV Music Profile but will vary in the degree of flexibility and conformance requirements that they embody.

### PROFILE IDENTIFIER

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

Music Profile Name	http://ns.osta.org/mpv/music/1.0/
--------------------	-----------------------------------

## EXAMPLE

```
<?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:mpvm="http://ns.osta.org/mpv/music/1.0/"
  xmlns:nmf="http://ns.osta.org/nmf/1.0/" >
  <nmf:Metadata>
    <ManifestProperties xmlns="http://ns.osta.org/manifest/1.0/">
      <ProfileBag>
        <Profile>http://ns.osta.org/mpv/basic/1.0/</Profile>
        <Profile>http://ns.osta.org/mpv/presentation/1.0/</Profile>
        <Profile>http://ns.osta.org/mpv/music/1.0/</Profile>
      </ProfileBag>
    </ManifestProperties>
  </nmf:Metadata>
  ...
</file:Manifest>
```

# Chapter 3: MPV Music Schema Introduction

## 3.1 Introduction

The MPV Music Profile makes use of the existing MPV Core specification and Basic and Presentation Profiles for creating collections of music and organizing them into albums / playlists. The MPV Music Profile augments this framework with additional metadata and practices specific to music.

The Music Profile provides a basic set of metadata which represents data and conventions used by software applications that create and play audio music on PCs or consumer electronics devices and music publishers of music CDs. The MultiAudio specification, already developed by OSTA [**MultiAudio**], is an earlier generation of technology; implementers are encouraged to utilize MPV. The music metadata that may be represented using the MPV Music Profile includes the following:

**Music Asset (“Song”, “Track”):** Asset Filename, Title, Principal artist, Album title, Genre, Playing time, Year recorded, Original order, Artwork, Music video, Performed by, Music by, Lyrics by, Arranged by, More info, Average encoded bitrate, Lyrics, Rights, Identifier, Description, Format,

**Album (“Playlist”) of Music:** Title, Principal Artist, Description, Identifier, Artwork, Music Entries

## 3.2 Examples

MPV Music playlists can range from simple to sophisticated, depending on the amount of available information and the ability of the creating application or device. Playback applications and devices determine the extent to which they use available information and the presentation of that information.

### 3.2.1 Namespaces and Profiles

All MPV files begin with a preamble that declares the XML namespaces and profiles used by the file. The `xmlns:xyz="namespace identifier"` sequence assigns a shortcut prefix (xyz) to represent the unique namespace identifier within the file. Use of namespaces allows the same element name to be used from different schema without ambiguity. For example, `<foo:Element>` and `<bar:Element>` are different if the namespace identifiers for each prefix are different and are the same if the namespace identifiers are the same.

A typical preamble:

```
<?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:mpvm="http://ns.osta.org/mpv/music/1.0/"
  xmlns:dc="http://ns.osta.org/nmf/1.0/dc/"
  xmlns:nmf="http://ns.osta.org/nmf/1.0/">
  <nmf:Metadata>
    <ManifestProperties xmlns="http://ns.osta.org/manifest/1.0/">
      <ProfileBag>
        <Profile>http://ns.osta.org/mpv/basic/1.0/</Profile>
        <Profile>http://ns.osta.org/mpv/presentation/1.0/</Profile>
        <Profile>http://ns.osta.org/mpv/music/1.0/</Profile>
      </ProfileBag>
    </ManifestProperties>
  </nmf:Metadata>
  ...

```

The Music profile in the manifest properties. As a best practice, the Basic profile also *SHOULD* be listed in the manifest. If an `<mpvp:Album>` element is provided, then the Presentation profile *SHOULD* be listed in the manifest. Adding the Basic and Presentation profile entries allow players that only understand those profiles to provide at least simple playback of a MPV file implementing the Music Profile.

### 3.2.2 Simple Example

This example MPV Music Profile file has 6 songs with only file location info for each item. There is no `<mpvp:Album>` playlist, so the sequence is the order of appearance in the `mpv:AssetList`. This example is the simplest form of using MPV for music playlists. Note that the file location of each song is provided in the two filesystems that typically occur on a CD, which are Joliet and ISO9660-1.

Even with this very simple usage, this MPV playlist adds value to the user's playback experience because the order of music playback is specified explicitly and is different from the sort order of the music by filename or file date.

```

<?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:mpvm="http://ns.osta.org/mpv/music/1.0/"
  xmlns:dc="http://ns.osta.org/nmf/1.0/dc/" xmlns:nmf="http://ns.osta.org/nmf/1.0/">
  <nmf:Metadata>
    <ManifestProperties xmlns="http://ns.osta.org/manifest/1.0/">
      <ProfileBag>
        <Profile>http://ns.osta.org/mpv/basic/1.0/</Profile>
        <Profile>http://ns.osta.org/mpv/music/1.0/</Profile>
      </ProfileBag>
    </ManifestProperties>
  </nmf:Metadata>
  <mpv:AssetList>
    <mpv:Audio mpv:id="01-GREAT-SWING-CLASSICS.MP3-20021202031833-a">
      <mpv:LastURL mpv:filesystem="Joliet">Great Swing Classics in HI-FI/Benny Goodman
      And His Orchestra - Jumpin' At The Woodside.mp3</mpv:LastURL>
      <mpv:LastURL mpv:filesystem="ISO9660-1">GREAT_SW/BENNY_GO.MP3</mpv:LastURL>
    </mpv:Audio>
    <mpv:Audio mpv:id="02-GREAT-SWING-CLASSICS.MP3-20021202031833-a">
      <mpv:LastURL mpv:filesystem="Joliet">Great Swing Classics in HI-FI/Duke Ellington
      And His Orchestra - Harlem Air Shaft.mp3</mpv:LastURL>
      <mpv:LastURL mpv:filesystem="ISO9660-1">GREAT_SW/DUKE_ELL.MP3</mpv:LastURL>
    </mpv:Audio>
    <mpv:Audio mpv:id="03-GREAT-SWING-CLASSICS.MP3-20021202031833-a">
      <mpv:LastURL mpv:filesystem="Joliet">Great Swing Classics in HI-FI/Stan Kenton And
      His Orchestra - Intermission Riff.mp3</mpv:LastURL>

```

```

    <mpv:LastURL mpv:filesystem="ISO9660-1">GREAT_SW/STAN_KEN.MP3</mpv:LastURL>
  </mpv:Audio>
  <mpv:Audio mpv:id="04-GREAT-SWING-CLASSICS.MP3-20021202031833-a">
    <mpv:LastURL mpv:filesystem="Joliet">Great Swing Classics in HI-FI/Harry James And
His Orchestra - I'm Beginning To See The Lig.mp3</mpv:LastURL>
    <mpv:LastURL mpv:filesystem="ISO9660-1">GREAT_SW/HARRY_J2.MP3</mpv:LastURL>
  </mpv:Audio>
  <mpv:Audio mpv:id="05-GREAT-SWING-CLASSICS.MP3-20021202031833-a">
    <mpv:LastURL mpv:filesystem="Joliet">Great Swing Classics in HI-FI/Glen Gray And
The Casa Loma Orchestra - Just An Old Manuscri.mp3</mpv:LastURL>
    <mpv:LastURL mpv:filesystem="ISO9660-1">GREAT_SW/GLEN_GR2.MP3</mpv:LastURL>
  </mpv:Audio>
  <mpv:Audio mpv:id="06-GREAT-SWING-CLASSICS.MP3-20021202031833-a">
    <mpv:LastURL mpv:filesystem="Joliet">Great Swing Classics in HI-FI/Les Brown And
His Orchestra - A Good Man Is Hard To Find.mp3</mpv:LastURL>
    <mpv:LastURL mpv:filesystem="ISO9660-1">GREAT_SW/LES_BROW.MP3</mpv:LastURL>
  </mpv:Audio>
</mpv:AssetList>
</file:Manifest>

```

### 3.2.3 Rich Example

In contrast to the previous example, this example of a MPV Music Profile file has much more information. In this case, two songs are specified along with a lot of information about the music including album artwork and music videos for the first song, and also a playlist (the *mpvp:Album* element) is provided that specified album/playlist-level information.

Careful reading of the contents of the *mpv:AssetList* in this example will show that not only the music songs but also still image and video assets are listed plus statements that relate the assets to each other. Not all these assets are considered “primary”, in other words, the user doesn’t want to interact with all assets equally. Primary assets are the ones that match the user’s idea of what the primary content is, such as a set of music songs. The *mpvp:Album* element is used to identify the primary assets, the sequence in which they should be presented, and other presentation information.

This example also illustrates how MPV Music Profile can be applied to a “hybrid” disc, such as a disc with that is both a DVD-Video disc and also contains MPV Music playlists and MP3 music of the songs. When played in a DVD-Video player, the user may enjoy watching the DVD-Video content, such as a music performance. In this case, no MPV Music information is used, just DVD-Video content and navigation. However, when played in a car stereo, only the MPV Music information is used and the player plays the MP3 music tracks that are also on the disc.

Some players will support both DVD-Video and MPV Music-based playback. In that case, for example, the MPV Music playback application may choose to allow the user to playback the associated music video for a track. In this example, the music video specified in the MPV playlist is actually the same music video played in DVD-Video mode, but it is accessed in a different way.

```

<?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:mpvm="http://ns.osta.org/mpv/music/1.0/"
  xmlns:dc="http://ns.osta.org/nmf/1.0/dc/"
  xmlns:nmf="http://ns.osta.org/nmf/1.0/">
  <nmf:Metadata>
    <ManifestProperties xmlns="http://ns.osta.org/manifest/1.0/">
      <ProfileBag>

```

```

    <Profile>http://ns.osta.org/mpv/basic/1.0/</Profile>
    <Profile>http://ns.osta.org/mpv/presentation/1.0/</Profile>
    <Profile>http://ns.osta.org/mpv/music/1.0/</Profile>
  </ProfileBag>
</ManifestProperties>
</nmf:Metadata>

<mpvp:Album> <!-- This defines an album / playlist presentation of the assets -->
  <nmf:Metadata> <!-- Album / playlist-level information -->
    <dc:Properties>
      <dc:description>14 swing classics re-recorded in the '50s by the original artists
for great sound with all the integrity and excitement of the original
performances.</dc:description>
      <dc:identifier>7243 5 21223 2 5 Capitol Jazz</dc:identifier>
      <dc:rights>(P) and (C) 1999 Capitol Records, Inc. All rights
reserved.</dc:rights>
      <dc:title>Music by Album and Track</dc:title>
    </dc:Properties>
    <mpvm:MusicProperties>
      <mpvm:AlbumTitle>Great SWING CLASSICS in HI-FI</mpvm:AlbumTitle>
      <mpvm:Genre>Jazz</mpvm:Genre>
      <mpvm:MoreInfoURL>www.bluenote.com</mpvm:MoreInfoURL>
    </mpvm:MusicProperties>
  </nmf:Metadata>
  <mpvp:Foreground> <!-- music playback sequence -->
    <mpv:AudioRef mpv:idRef="01-GREAT-SWING-CLASSICS.MP3-20021202031833-a"/>
    <mpv:AudioRef mpv:idRef="02-GREAT-SWING-CLASSICS.MP3-20021202031833-a"/>
  </mpvp:Foreground>
</mpvp:Album>

<mpv:AssetList> <!-- This is the per-asset info -->
  <mpv:Audio mpv:id="01-GREAT-SWING-CLASSICS.MP3-20021202031833-a">
    <mpv:ContentID>urn:osta-
org:mpv:dsig:md5:all:93446AEFA3B340DA971BAF09B17DBC394</mpv:ContentID>
    <mpv:LastURL mpv:filesystem="Joliet">Benny Goodman And His Orchestra - Jumpin' At The
Woodside.mp3</mpv:LastURL>
    <mpv:LastURL mpv:filesystem="ISO9660-1">BENNY_GO.MP3</mpv:LastURL>
    <nmf:Metadata>
      <dc:Properties>
        <dc:creator>Benny Goodman and his Orchestra</dc:creator>
        <dc:description/>
        <dc:format>audio/mpeg</dc:format>
        <dc:identifier/>
        <dc:title>Jumpin' At The Woodside</dc:title>
      </dc:Properties>
      <mpvm:MusicProperties>
        <mpvm:AlbumTitle>Great SWING CLASSICS in HI-FI</mpvm:AlbumTitle>
        <mpvm:ArrangedBy>Count Basie;Jimmy Mundy</mpvm:ArrangedBy>
        <mpvm:Genre>Jazz</mpvm:Genre>
        <mpvm:MusicBy>Count Basie</mpvm:MusicBy>
        <mpvm:OrigIndex>1</mpvm:OrigIndex>
        <mpvm:PlayingTime>00:03:28</mpvm:PlayingTime>
        <mpvm:PrincipalArtist>Benny Goodman</mpvm:PrincipalArtist>
        <mpvm:Recorded>1954-11-09</mpvm:Recorded>
      </mpvm:MusicProperties>
    </nmf:Metadata>
    <mpv:Related mpv:relationship="urn:osta-org:mpv:music:artwork">
      <mpv:StillRef mpv:idRef="01-GREAT-SWING-CLASSICS.MP3-20021202031833-b"/>

```

```

</mpv:Related>
<mpv:Related mpv:relationship="urn:osta-org:mpv:music:video">
  <mpv:VideoRef mpv:idRef="01-GREAT-SWING-CLASSICS.MP3-20021202031833-c"/>
</mpv:Related>
</mpv:Audio>

<!-- artwork for the first song -->
<mpv:Still mpv:id="01-GREAT-SWING-CLASSICS.MP3-20021202031833-b">
  <mpv:LastURL mpv:filesystem="Joliet">Artwork/Benny Goodman And His
Orchestra.jpg</mpv:LastURL>
  <mpv:LastURL mpv:filesystem="ISO9660-1">ARTWORK/BENNY_GO.JPG</mpv:LastURL>
  <nmf:Metadata>
    <dc:Properties>
      <dc:format>image/jpeg</dc:format>
    </dc:Properties>
  </nmf:Metadata>
</mpv:Still>

<!-- music video for the first song -->
<mpv:Video mpv:id="01-GREAT-SWING-CLASSICS.MP3-20021202031833-c">
  <!-- note use of a video segment of a DVD-Video disc. This won't play on a PC unless
it is unencrypted -->
  <mpv:LastURL>VIDEO_TS/VTS_01_1.VOB</mpv:LastURL>
  <nmf:Metadata>
    <dc:Properties>
      <dc:format>video/mpeg</dc:format>
    </dc:Properties>
  </nmf:Metadata>
</mpv:Video>

<mpv:Audio mpv:id="02-GREAT-SWING-CLASSICS.MP3-20021202031833-a">
  <mpv:ContentID>urn:osta-
org:mpv:dsig:md5:all:EF886AEFA3B340DA971BAF09B17DBC122</mpv:ContentID>
  <mpv:LastURL mpv:filesystem="Joliet">Duke Ellington And His Orchestra - Harlem Air
Shaft.mp3</mpv:LastURL>
  <mpv:LastURL mpv:filesystem="ISO9660-1">DUKE_ELL.MP3</mpv:LastURL>
  <nmf:Metadata>
    <dc:Properties>
      <dc:creator>Duke Ellington and his Orchestra</dc:creator>
      <dc:description/>
      <dc:format>audio/mpeg</dc:format>
      <dc:identifier/>
      <dc:title>Harlem Air Shaft</dc:title>
    </dc:Properties>
    <mpvm:MusicProperties>
      <mpvm:AlbumTitle>Great SWING CLASSICS in HI-FI</mpvm:AlbumTitle>
      <mpvm:PrincipalArtist>Duke Ellington</mpvm:PrincipalArtist>
      <mpvm:MusicBy>Duke Ellington</mpvm:MusicBy>
      <mpvm:AlbumTitle>Great SWING CLASSICS in HI-FI</mpvm:AlbumTitle>
      <mpvm:Recorded>1955-11-17</mpvm:Recorded>
      <mpvm:Genre>Jazz</mpvm:Genre>
      <mpvm:OrigIndex>2</mpvm:OrigIndex>
      <mpvm:PlayingTime>00:03:54</mpvm:PlayingTime>
    </mpvm:MusicProperties>
  </nmf:Metadata>
</mpv:Audio>
</mpv:AssetList>
</file:Manifest>

```

Filenames of assets are specified using the *mpv:LastURL* element. Pathnames can be relative or absolute; relative names begin relative to the location of the MPV file. Pathnames are to be specified using URL-compliant syntax. This includes translation of special characters like the space (“ ”) into equivalent representations like “%20” and use of prefixes like *file:///* to introduce absolute pathnames to local files. Multiple pathnames may be specified for any given asset; they are interpreted as alternate paths to the same set of bits. A processing application tries them sequentially to try and locate the asset.

The term “LastURL” is used to emphasize that it’s value is a URL to the last-known location of the file; because media files may be moved or renamed independently of the MPV file, it is possible that the media file has moved and must be searched for. The *mpv:InstanceID* and *mpv:ContentID* elements, if provided, are identifiers that can be used to find files that cannot be located by any of the *LastURL* entries.

### 3.3 Use of Existing MPV Specifications

The MPV Music Profile uses MPV in a manner consistent with these the existing MPV Core specification and the MPV Basic Profile and Presentation Profile specifications. For metadata, it incorporates the MPV Dublin Core NMF specification for those properties that can be represented in that manner.

### 3.4 MPV Music Profile Metadata Introduction

The MPV Core specification already supports an *mpv:Audio* asset type. The MPV Presentation Profile specification describes how to create playlists (“*mpvp:Album*”) of assets, such as music and images. To this framework, the MPV Music Profile adds extensive metadata specifically about music; the existing framework continues to be used in a manner fully consistent with existing specifications.

<i>Metadata</i>	<i>MPV Music Profile</i>	<i>Discussion</i>
<b>Music Asset (“Song”, “Track”)</b>	<b>Subelements of the <i>mpv:Audio</i> asset</b>	
Pathname	<i>mpv:LastURL</i>	one or more pathnames that should resolve to the music file
Title	<i>nmf:Metadata dc:Properties dc:title</i>	the music “title”
Genre	<i>nmf:Metadata mpvm:MusicProperties mpvm:Genre</i>	Predefined set of genres can be extended simply by using a new name.
Principal artist	<i>nmf:Metadata mpvm:MusicProperties mpvm:PrincipalArtist</i>	
Album Title	<i>nmf:Metadata mpvm:MusicProperties mpvm:AlbumTitle</i>	
Date Recorded	<i>nmf:Metadata dcterms:Properties dcterms:created</i>	recorded date-time, e.g. 2002-10-03T21:07:00Z
Identifier	<i>nmf:Metadata dc:Properties dc:identifier</i>	Catalog number, UPC, etc. A text string, not classified in any way and not guaranteed unique.



Description	nmf:Metadata dc:Properties dc:description	Commentary, message, description
Format	nmf:Metadata dc:Properties dc:format	use MPV-provided list of well-known MIME types or vendor-specific extended types
Rendition using another codec	mpv:Rendition mpv:renditionType="alt"  mpv:AudioRef	Refers to an alternate codec encoding of the asset.
Rendition with another bitrate	mpv:Rendition mpv:renditionType="subsampld"  mpv:AudioRef	Refers to another encoding of the asset at a different bitrate. Use mpvm:AvgEncodedBitrate or interrogate the file for more information.
Asset artwork	mpv:Related mpv:relationship="urn:osta-org:mpv:music:artwork"  mpv:StillRef	Refers to an image or other content that is artwork related to the asset. The referenced asset does not have to be a Still, it can be any asset type. The "dc:title" of the related asset can be used to describe the artwork.
Music video	mpv:Related mpv:relationship="urn:osta-org:mpv:music:video"  mpv:VideoRef	Refers to a video that is of the same music as the primary music track.
Performed by	nmf:Metadata dc:Properties dc:creator	Performer names in text
Music by	nmf:Metadata mpvm:MusicProperties mpvm:MusicBy	ComposerName
Lyrics by	nmf:Metadata mpvm:MusicProperties mpvm:LyricsBy	SongwriterName
Arranged by	nmf:Metadata mpvm:MusicProperties mpvm:ArrangedBy	ArrangerName
Online Info	nmf:Metadata mpvm:MusicProperties mpvm:MoreInfoURL	URL to follow to get more info about the music.
Playing Time	nmf:Metadata mpvm:MusicProperties mpvm:PlayingTime	in seconds, duration
Track Order	nmf:Metadata mpvm:MusicProperties mpvm:OrigIndex	sequence order of the audio track on the original media, such as an audio CD. Starts with 1, not 0.
Encoded Bitrate	nmf:Metadata mpvm:MusicProperties mpvm:EncodedBitrate	Maximum bitrate in bps of the encoded asset. For variable bitrate-encoded assets, this is the maximum value; for constant bitrate encoded assets, it is the average value.
Lyrics	nmf:Metadata mpvm:MusicProperties mpvm:Lyrics	Time offset, text, language??
Extra Data	nmf:Metadata and mpv:Metadata	Any metadata that an application whichs to create can go into these containers.
Key-Value Pairs	nmf:Metadata mpvm:Properties mpvm:KeyValue mpvmkv:Key nmf:Metadata mpvm:Properties mpvm:KeyValue mpvmkv:Value	Arbitrary key-value pairs
<b>Mood</b>	<b>nmf:Metadata mpvm:MusicProperties mpvm:Mood</b>	<b>What mood is this music?</b>

		(e.g. mellow, wild)
Tempo	<code>nmf:Metadata mpvm:MusicProperties mpvm:Tempo</code>	What tempo is this music? (e.g. fast, slow)
Situation	<code>nmf:Metadata mpvm:MusicProperties mpvm:Situation</code>	What situation is this music for? (e.g dance, dinner)

<b>Album (“Playlist”) of Music</b>	<b>mpvp:Album</b>	
Number of Entries	implicit – number of entries in the foreground	
Title	<code>nmf:Metadata dc:Properties dc:title</code>	
Performed By	<code>nmf:Metadata dc:Properties dc:creator</code>	Performer names
Principal Artist	<code>nmf:Metadata mpvm:MusicProperties mpvm:PrincipalArtist</code>	Principal artist’s name
Description	<code>nmf:Metadata dc:Properties dc:description</code>	
Identifier	<code>nmf:Metadata dc:Properties dc:identifier</code>	catalog number, IDC
Album Artwork	<code>mpvp:Album mpv:Related mpv:relationship=”urn:osta-org:mpv:music:artwork” mpv:StillRef</code>	
Music Playlist Items	<code>mpvp:Album mpvp:Foreground</code>	This content is played as the principal content of the music playlist.
Playlist accompaniment	<code>mpvp:Album mpvp:Background</code>	This content is played as background to the music content in the foreground. For example, it could specify images.
Key-Value Pairs	<code>nmf:Metadata mpvm:Properties mpvm:KeyValue mpvmkv:Key</code> <code>nmf:Metadata mpvm:Properties mpvm:KeyValue mpvmkv:Value</code>	Arbitrary key-value pairs
Extra Data	<code>mpv:Metadata</code> and <code>nmf:Metadata</code> anywhere	

### 3.5 Use of Dublin Core Metadata

To promote interoperability, MPV makes use of the Dublin Core metadata [DC-NMF] to represent essential metadata across all types of assets. Thus, the `dc:title` element specifies the title of a music asset just the same as an image or video asset.

In the previous section, the Dublin Core metadata elements were mixed into the overall set of music metadata properties and assets. To clarify usage of the DC metadata, this section extracts just the Dublin Core elements from the previous section and groups them together for convenience.

<b>Music Asset (“Song”, “Track”)</b>	<b>mpv:Audio subelements</b>	
Performed by	<code>nmf:Metadata dc:Properties dc:creator</code>	Performer Names
Description	<code>nmf:Metadata dc:Properties dc:description</code>	Commentary, message, description
Format	<code>nmf:Metadata dc:Properties dc:format</code>	use MPV-provided list of well-known MIME types

Identifier	nmf:Metadata dc:Properties dc:identifier	Catalog number, UPC, etc. A text string, not classified in any way and not guaranteed unique.
Title	nmf:Metadata dc:Properties dc:title	Free-form plain text
Created Date	nmf:Metadata dcterms:Properties dcterms:created	date-time music performed or recorded

<b>Album (“Playlist”) of Music</b>	<b>mpvp:Album subelements</b>	
Performed By	nmf:Metadata dc:Properties dc:creator	
Description	nmf:Metadata dc:Properties dc:description	
Identifier	nmf:Metadata dc:Properties dc:identifier	catalog number, IDC
Title	nmf:Metadata dc:Properties dc:title	

### 3.6 Music Media Types

MPV is an open format that can support an expandable set of defined formats. Formats are identified using MIME media types, as is well-established practice for internet-era standards.

The [MPV-Core] specification defines the following music audio formats.

#### MEDIA TYPES FOR MPV: AUDIO FROM [MPV-CORE]

<u>MIME Media Type</u>	<u>Mac File Type</u>	<u>PC File Suffixes</u>	<u>Description</u>
audio/basic	ULAW	au, snd	8K, mono audio
audio/midi	MIDI	mid, midi	Musical Instrument Digital Interface sound file
audio/mpeg	MPEG	mp1, mp2, mp3	MPEG1 audio layers 1, 2, and 3
audio/wav	WAVE	wav	WAVE file
audio/x-aiff	AIFF	aif, aiff	Audio interchange file format
audio/x-ms-wma		wma	Windows Media Audio

This MPV Music Profile specification defines the following additional music audio formats.

#### ADDITIONAL MEDIA TYPES FOR MPV: AUDIO

<u>MIME Media Type</u>	<u>Mac File Type</u>	<u>PC File Suffixes</u>	<u>Description</u>
audio/ac3		.ac3	Dolby Digital 5.1 audio used widely for DVD-Video
audio/MP4A-LATM		.aac	Audio using MPEG4 audio encoding.
		.omg	Sony’s ATRAC3 format

# Chapter 4: MPV Music Profile – Using MPV Playlists

## 4.1 MPV Music Playlists

MPV playlists are a central concept in MPV that provide for user-friendly organization and navigation of the music, photo, and video (and other) content on a storage media. A MPV playlist identifies the primary assets to be presented to the user, such as a set of music songs. The playlist also specifies the sequence in which assets should be presented and other presentation characteristics.

MPV playlists are implemented according to the MPV Presentation Profile specification [MPV-Pres]. This MPV Music Profile document specifies the details of this usage. The basic structure of an MPV playlist uses the <mpvp:Album> element, which may contain metadata, foreground and background content, and related and rendition assets.

```
...
<mpvp:Album>
  <mpv:Metadata> ... </mpv:Metadata>
  <nmf:Metadata> ... </nmf:Metadata>
  <mpvp:Background> ... </mpvp:Background>
  <mpvp:Foreground> ... </mpvp:Foreground>
  <mpv:Rendition> ... </mpv:Rendition>
  <mpv:Related> ... </mpv:Related>
</mpvp:Album>
...
```

MPV is focused on interoperability of content produced and consumed on both PCs and consumer electronics devices. Unlike CE-focused solutions, it is expected that users will organize media files into folders/directories of their own choosing and give files long filenames. This means that it isn't possible to specify content location and playback order simply by requiring specific directory and filenames. Instead, MPV provides an approach that allows the content to be located anywhere and with any name.

In addition, many related media assets may go onto a storage media such as CD that are used to enhance access and playback performance and to provide enhanced playback and printing experiences. The MPV playlist allows this content to be managed to present the user a simple high-level interaction with their content.

### 4.1.1 One Playlist per MPV File

A MPV file conforming to the MPV Music Profile 1.0 MUST contain only a single playlist; in other words, it contains only a single <mpvp:Album> element. In most cases, the MPV file SHOULD contain in the

<mpv:AssetList> the assets that are referenced by the <mpvp:Album>'s contents. While the asset references have the ability to reference MPV assets located in another MPV manifest file, this usage is not recommended by the MPV Music Profile because it requires special handling by the MPV reader.

### 4.1.2 Metadata Usage

Metadata placed on the mpvp:Album element has the scope of the whole album or playlist. For example, the mpvm:PrincipalArtist element can be used with either a single music asset or on the mpvp:Album.

<b>Album (“Playlist”) of Music</b>	<b>mpvp:Album</b>	
Number of Entries	implicit – number of entries in the foreground	
Title	nmf:Metadata dc:Properties dc:title	
Performed By	nmf:Metadata dc:Properties dc:creator	Performer names
Principal Artist	nmf:Metadata mpvm:MusicProperties mpvm:PrincipalArtist	Principal artist’s name
Description	nmf:Metadata dc:Properties dc:description	
Identifier	nmf:Metadata dc:Properties dc:identifier	catalog number, IDC
Album Artwork	mpvp:Album mpv:Related mpv:relationship=”urn:osta-org:mpv:music:artwork” mpv:StillRef	
Music Playlist Items	mpvp:Album mpvp:Foreground	This content is played as the principal content of the music playlist.
Playlist accompaniment	mpvp:Album mpvp:Background	This content is played as background to the music content in the foreground. For example, it could specify images.
Key-Value Pairs	nmf:Metadata mpvm:Properties mpvm:Key Value mpvmkv:Key nmf:Metadata mpvm:Properties mpvm:Key Value mpvmkv:Value	Arbitrary key-value pairs
Extra Data	mpv:Metadata and nmf:Metadata anywhere	

### 4.1.3 Background Usage

The mpvp:Background element is used in the MPV Music Profile to provide background presentation content to the music in the foreground. It is applied at the discretion of the playback application or device. If it is a simple device or application with no visual presentation capabilities, the mpvp:Background content SHOULD be ignored.

If the application or device has visual presentation capabilities, the mpvp:Background content may be processed. The recommended content is a series of mpv:StillRef elements that refer to still images that may be displayed behind the foreground content. This can be used in both browse/menu mode as well as playback mode. All the [MPV-Pres] properties may be applied, such as transitions. These are honored at the discretion of the playback application.

### 4.1.4 Foreground Usage

The MPV music playlist contains a foreground list (*mpvp:Foreground*) of music assets identified using the *mpv:AudioRef* element. This defines the set of primary music items in the playlist and their sequence. In addition to the referenced audio assets, the mpv:AssetList in the MPV file may also contain still image, video, and other kinds of assets. The *mpvp:Album* makes it easy to identify which assets in the *mpv:AssetList* are primary by explicitly identifying them.

## 4.1.5 Related and Rendition Assets

A *mpvp:Album* may identify renditions and related assets. Album artwork may be specified using a *mpv:Related* element with an *urn:osta-org:mpv:music:artwork* relationship identifier. A pre-generated playback video of the whole album/playlist may be specified using a *mpv:Rendition* element with a *mpv:renditionUsage*="show".

## 4.2 Linked Playlists

The [MPV-Core] specification established the structure and nomenclature of MPV files and assets. The MPV file is an XML document that is called an MPV Manifest and the outer-most element of a MPV file is a `<file:Manifest>`.

One very useful capability that [MPV-Core] provides is to link manifests to one another. The `<mpv:ManifestLink>` element creates a link to another MPV file. In this manner, just as with the WorldWideWeb, an endless chain of linked MPV files can be created.

Typically, when applied to a removable storage media like a CD, DVD, or memory card, all the links will be self-contained within the media. In this case, typically a file using the distinguished filename "index.pvm" will contain a top-level list of linked playlists.

### EXAMPLE:

In this example, a playlist consists only of links to three other playlists. Each playlist link has a screen-resolution image representation that can be used to enhance to a graphical presentation. Note that the filesystem location and names of these playlists is arbitrary and may be located at the choice of the authoring application or device. Inspection of the linked playlists shows that they are specific orderings of the music – by genre, by artist, and by album. Thus the example is typical of a top-level playlist (e.g. an index.pvm) which provides for access to other playlists with specific content organization. Each of the specific playlists could be further organized into additional playlists, e.g. one for each genre, artist, and 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:mpvm="http://ns.osta.org/mpv/music/1.0/" xmlns:nmf="http://ns.osta.org/nmf/1.0/">
  <nmf:Metadata>
    <ManifestProperties xmlns="http://ns.osta.org/manifest/1.0/">
      <InstanceID>23686AEFA3B340DAAC1BAF09B17DBCBC9</InstanceID>
      <ProfileBag>
        <Profile>http://ns.osta.org/mpv/basic/1.0/</Profile>
        <Profile>http://ns.osta.org/mpv/music/1.0/</Profile>
        <Profile>http://ns.osta.org/mpv/presentation/1.0/</Profile>
      </ProfileBag>
    </ManifestProperties>
  </nmf:Metadata>

  <mpvp:Album mpv:id="ALB001">
    <nmf:Metadata>
      <Properties xmlns="http://purl.org/dc/elements/1.1/">
        <creator>Pieter van Zee</creator>
        <description>A collection of my favorites songs from the 40's and
50's</description>
        <title>Golden Oldies party music</title>
      </Properties>
      <Properties xmlns="http://purl.org/dc/terms/1.1/">
```

```

        <created>2002-12-01T23:11:00Z</created>
    </Properties>
</nmf:Metadata>
<mpvp:Background>
    <mpv:StillRef mpv:idRef="ID000000"/>
</mpvp:Background>
<mpvp:Foreground>
    <mpv:ManifestLinkRef mpv:idRef="ID000100"/>
    <mpv:ManifestLinkRef mpv:idRef="ID000200"/>
    <mpv:ManifestLinkRef mpv:idRef="ID000300"/>
</mpvp:Foreground>
</mpvp:Album>

<mpv:AssetList>
    <!-- background image for the playlist presentation -->
    <mpv:Still mpv:id="ID000000">
        <mpv:LastURL mpv:filesystem="Joliet">Playlists/Artwork/index.jpg</mpv:LastURL>
        <mpv:LastURL mpv:filesystem="ISO9660-1">PLAYLIST/ARTWORK/INDEX.JPG</mpv:LastURL>
    </mpv:Still>

    <!-- ManifestLink -->
    <mpv:ManifestLink mpv:id="ID000100">
        <mpv:InstanceID>EF886AEFA3B340DA971BAF09B17DBC122</mpv:InstanceID>
        <mpv:LastURL mpv:filesystem="Joliet">Playlists/by genre.pvm</mpv:LastURL>
        <mpv:LastURL mpv:filesystem="ISO9660-1">PLAYLIST/BY_GENRE.PVM</mpv:LastURL>
        <nmf:Metadata>
            <Properties xmlns="http://purl.org/dc/elements/1.1/">
                <title>By Genre</title>
            </Properties>
        </nmf:Metadata>
        <mpv:Rendition mpv:renditionUsage="screen">
            <mpv:StillRef mpv:idRef="ID000101"/>
        </mpv:Rendition>
    </mpv:ManifestLink>
    <!-- thumbnail stand-in for the manifest -->
    <mpv:Still mpv:id="ID000101">
        <mpv:LastURL mpv:filesystem="Joliet">Playlists/Artwork/by genre.jpg</mpv:LastURL>
        <mpv:LastURL mpv:filesystem="ISO9660-1">PLAYLIST/ARTWORK/BY_GENRE.JPG</mpv:LastURL>
    </mpv:Still>

    <!-- ManifestLink -->
    <mpv:ManifestLink mpv:id="ID000200">
        <mpv:InstanceID>34FE6AEFA3B340DA891BAF09B17DBC992</mpv:InstanceID>
        <mpv:LastURL mpv:filesystem="Joliet">Playlists/by artist.pvm</mpv:LastURL>
        <mpv:LastURL mpv:filesystem="ISO9660-1">PLAYLIST/BY_ARTIS.PVM</mpv:LastURL>
        <nmf:Metadata>
            <Properties xmlns="http://purl.org/dc/elements/1.1/">
                <title>By Artist</title>
            </Properties>
        </nmf:Metadata>
        <mpv:Rendition mpv:renditionUsage="screen">
            <mpv:StillRef mpv:idRef="ID000201"/>
        </mpv:Rendition>
    </mpv:ManifestLink>
    <!-- thumbnail stand-in for the manifest -->
    <mpv:Still mpv:id="ID000201">
        <mpv:LastURL mpv:filesystem="Joliet">Playlists/Artwork/by artist.jpg</mpv:LastURL>
        <mpv:LastURL mpv:filesystem="ISO9660-1">PLAYLIST/ARTWORK/BY_ARTIS.JPG</mpv:LastURL>
    </mpv:Still>

```

```

</mpv:Still>
<!-- ManifestLink -->
<mpv:ManifestLink mpv:id="ID000300">
  <mpv:InstanceID>93486AEFA3B340DA231BAF09B17DBCEFB</mpv:InstanceID>
  <mpv:LastURL mpv:filesystem="Joliet">Playlists/by album.pvm</mpv:LastURL>
  <mpv:LastURL mpv:filesystem="ISO9660-1">PLAYLIST/BY_ALBUM.PVM</mpv:LastURL>
  <nmf:Metadata>
    <Properties xmlns="http://purl.org/dc/elements/1.1/">
      <title>By Album</title>
    </Properties>
  </nmf:Metadata>
  <mpv:Rendition mpv:renditionUsage="screen">
    <mpv:StillRef mpv:idRef="ID000301"/>
  </mpv:Rendition>
</mpv:ManifestLink>
<!-- thumbnail stand-in for the manifest -->
<mpv:Still mpv:id="ID000301">
  <mpv:LastURL mpv:filesystem="Joliet">Playlists/Artwork/by album.jpg</mpv:LastURL>
  <mpv:LastURL mpv:filesystem="ISO9660-1">PLAYLIST/ARTWORK/BY_ALBUM.JPG</mpv:LastURL>
</mpv:Still>
</mpv:AssetList>
</file:Manifest>

```

This example would exist in the context of the following sample filesystem organization on a CD.

```

/index.pvm
/Playlists/by album.pvm
/Playlists/by artist.pvm
/Playlists/by genre.pvm
/Playlists/Artwork/by album.jpg
/Playlists/Artwork/by artist.jpg
/Playlists/Artwork/by genre.jpg
/Playlists/Artwork/index.jpg

```

## 4.3 Default Playlists

Each MPV file represents exactly one playlist, so there is no need to have a default playlist specified within the file. Instead, the default playlist behaviour is determined by the order in which playlists are processed by the application or device that is accessing them.

The default playlist is the first playlist that is encountered. Because [MPV-Core] fixed the algorithm by which MPV files are searched for, this is deterministic. *Index.pvm* is the first file searched for, followed by *indexpvm.xml*, *album.pvm*, and *albumpvm.xml*. Then files with the *.pvm* extension are processed in alphabetical order.

How the default playlist is processed is up to the processing application or device. The recommended best practices are that when the user is “browsing” or viewing a menu, the contents of the default list SHOULD be presented to the user. When the user is “playing/showing”, the contents of the list SHOULD be recursively played or interpreted. When a manifestLink is played, the behaviour is to open the file referenced by the link and play it.

For example, in “browse / menu” mode, the example file above would be presented to the user as a menu from which he could choose which playlist to use next. When in “playback” mode, the “By genre.pvm” MPV file would be opened and it would be played.



## 4.4 Dynamic Playlists

So far, we have described how to create playlists of fixed collections of music. However, it is valuable to be able to represent the set of selection criteria that produced a given collection. This allows for round-trip editing and update of playlists and music collections when applied to an evolving collection of music over time.

Dynamic playlists also allow the more advanced player applications and devices to dynamically extend the playlists to reflect the music that is currently actually available. This is particularly valuable for hard-drive applications of MPV music playlists, where the user is regularly adding more music to a collection.

Imagine, for example, a harddrive-based music jukebox that contained your ever-growing complete collection of music. As you add new CDs to the collection, your jukebox playlists are automatically updated to include the new music selections. In addition, each time you insert a disc of music into the music jukebox, the MPV-aware jukebox offers to incrementally add more music to the disc from its collection that met the specified selection criteria of the playlists on the disc.

Support for dynamic playlists MAY be implemented in a maker or player application or device.

### 4.4.1 Specifying Dynamic Playlists

Dynamic playlists are specified essentially like database queries. They are stored as metadata on the `mpvp:Foreground` and `mpvp:Background` elements of an `mpvp:Album`. A processing application would apply them to the set of assets listed in the `mpv:AssetList`. Of course, this list may shrink or grow as more or less music is added to it.

Dynamic playlists can be specified with or without a set of static asset references. For the MPV Music Profile 1.0, a MPV playlist on removable storage media MUST include static asset references; it MAY include dynamic playlist queries that represent the selection criteria for the static assets that are specified.

### 4.4.2 Playlist Query Language

MPV Music Profile 1.0 allows applications to specify XPath queries that result in a list of assets that represent a MPV playlist. Since MPV is an XML-based grammar, it can use the widely adopted grammar called XPath [XPath] that provides an industry-standard means to express selection criteria for one or more elements of an XML document.

Software SDKs that process XPath queries are widely available for most PC and server computing platforms and several multiplatform implementations are available. Low-end platforms, however, may not wish to support XPath queries because of the firmware and runtime memory requirements.

The XPath expression is executed relative to the `<mpv:AssetList>` node in the target MPV manifest file. The resulting list of assets that match the statement are treated for that instance as members of the playlist.

Todo:  
Metadata grammar  
Example

## 4.5 Best Practices for Generating Playlists

There are any number of ways for applications and devices to produce playlists. They may be the result of extensive manual user interaction. Or they may be generated automatically based on music metadata such as Principal Artist, Genre, and Album Title.

### **4.5.1 Top-Level Playlist**

The most important best practice is to provide a top-level playlist, such as an `index.pvm`. This gives the playback application or device the best ability to navigate contents in the order intended by the creator application.

### **4.5.2 “All Music” Playlist**

It is RECOMMENDED that at least one playlist on a storage media play all the music. If this playlist is also the first playlist in the top-level playlist, it is the playlist that gets played first when a default playlist is used.

### **4.5.3 Local mpv:AssetList**

Thus far, we have described putting the `mpv:AssetList` in each MPV playlist file. Putting the assets in a local assetlist duplicates the asset data but simplifies reading the playlist because all the information is contained in one location and data not used by the playlist can be left out. A removable storage media that implements MPV Music Profile 1.0 playlists MUST put the asset data in the `mpv:AssetList` of the same MPV file as the `mpvp:Album`.

### **4.5.4 Centralized mpv:AssetList**

However, a jukebox device such as the one described in Section 4.4, Dynamic Playlists, might store the `mpv:AssetList` representing the full set of music assets in a separate MPV file and create separate playlists expressed using dynamic playlists. This approach is efficient and scalable because it avoids duplicating information about the assets in more than one place. In addition to dynamic playlists, this approach can also accommodate static playlists by using the `mpv:manifestLinkIdRef` attribute in the `mpv:AudioRef` asset reference in the `mpvp:Foreground` or `Background`.

Usage of a centralized assetlist is defined by MPV Core and is quite practical for many applications, but it is not supported by the MPV Music Profile 1.0 specification for use on removable storage media because it requires a somewhat more capable playback device or application. It is suitable for management of assets on non-removable media, such as a computer or device harddrive.

## Chapter 5: MPV Music Schemas In Detail

### 5.1 Multiple Renditions of a Music Asset

The file specified in the primary asset should generally be the highest quality encoding while maintaining high compatibility with typical players. It may be useful to specify multiple renditions of a music asset. For example, if space is available on the storage media, multiple encodings of the same music asset can be specified, increasing the compatibility of the music disc with multiple players that may have different supported codecs and sustained throughputs.

Renditions of a music asset are specified using the <mpv:Rendition> tag and appropriate mpv:renditionUsage attribute values. For music asset renditions based on subsampling, the “subsampled” value of the mpv:renditionUsage attribute is the most appropriate. In addition, there may be multiple codec encodings of a given asset. These are encoding using the “alt” value of the mpv:renditionUsage attribute.

RenditionUsage	Description
alt	Alternate codec encoding of the same music asset. Metadata on the asset that is referenced can indicate the codec (nmf:Metadata dc:Properties dc:format) and bitrate (nmf:Metadata mpvm:Properties mpvm:EncodedBitrate).
subsampled	An encoding in the same codec as the primary asset but with a lesser bitrate. Metadata on the asset that is referenced can indicate the bitrate (nmf:Metadata mpvm:Properties mpvm:EncodedBitrate).
mpvm:EncodedBitrate	<p>Describes the throughput in bits-per-second of the content as an integer value. For variable-bitrate-encoded assets, use the maximum value. For constant-bitrate-encoded assets, use the constant value.</p> <p>Typical values are 64000, 96000, 128000, 192000, 256000, 320000</p> <p>Note that these values are not Kbps values, where K=1024, but kbps values where k=1000.</p>

**EXAMPLE**

In this example, a primary MP3 file encoded at 256kbps has a MP3 file rendition encoded at 64kbps and a WMA file alternate rendition encoded at 64kbps.

```

...
<mpv:AssetList>
  <!-- This is the per-asset info -->
  <mpv:Audio mpv:id="01-GREAT-SWING-CLASSICS.MP3-20021202031833-a">
    <mpv:LastURL>01 Great Swing Classics.mp3</mpv:LastURL>
    <nmf:Metadata>
      <dc:Properties>
        <dc:creator>Benny Goodman and his Orchestra</dc:creator>
        <dc:format>audio/mpeg</dc:format>
        <dc:title>Jumpin' At The Woodside</dc:title>
      </dc:Properties>
      <mpvm:MusicProperties>
        <mpvm:AlbumTitle>Great SWING CLASSICS in HI-FI</mpvm:AlbumTitle>
        <mpvm:Genre>Jazz</mpvm:Genre>
        <mpvm:EncodedBitrate>256000</mpvm:EncodedBitrate>
      </mpvm:MusicProperties>
    </nmf:Metadata>
    <mpv:Rendition mpv:renditionUsage="subsampled">
      <mpv:AudioRef mpv:idRef="01-GREAT-SWING-CLASSICS.MP3-20021202031833-R1"/>
    </mpv:Related>
    <mpv:Related mpv:relationship="alt">
      <mpv:AudioRef mpv:idRef="01-GREAT-SWING-CLASSICS.MP3-20021202031833-R2"/>
    </mpv:Related>
  </mpv:Audio>
  <mpv:Audio mpv:id="01-GREAT-SWING-CLASSICS.MP3-20021202031833-R1">
    <mpv:LastURL>01 Great Swing Classics.mp3</mpv:LastURL>
    <nmf:Metadata>
      <dc:Properties>
        <dc:format>audio/mpeg</dc:format>
      </dc:Properties>
      <mpvm:MusicProperties>
        <mpvm:EncodedBitrate>64000</mpvm:EncodedBitrate>
      </mpvm:MusicProperties>
    </nmf:Metadata>
  </mpv:Audio>
  <mpv:Audio mpv:id="01-GREAT-SWING-CLASSICS.MP3-20021202031833-R2">
    <mpv:LastURL>01 Great Swing Classics.wma</mpv:LastURL>
    <nmf:Metadata>
      <dc:Properties>
        <dc:format>audio/x-ms-wma</dc:format>
      </dc:Properties>
      <mpvm:MusicProperties>
        <mpvm:EncodedBitrate>64000</mpvm:EncodedBitrate>
      </mpvm:MusicProperties>
    </nmf:Metadata>
  </mpv:Audio>

```

## 5.2 Artwork for an Asset

The MPV Music Profile allows rich types of artwork for an asset to be specified. A generating application can specify multiple kinds of artwork with any given music asset. The generating application has a choice: it can use the generic “relationship” string of “urn:osta-org:mpv:music:artwork”, or if available, a more specific relationship may be specified.

A processing application that doesn’t care about specific types of artwork can do a string match against the more generic “urn:osta-org:mpv:music:artwork” relationship, while more specific needs can also be matched. Note also that multiple related assets can be identified that refer to the same physical asset. For example, the cover artwork may also be artwork of a performance.

<i>Relationship type on an asset</i>	<i>Meaning</i>
urn:osta-org:mpv:music:artwork	Unspecified artwork.
urn:osta-org:mpv:music:artwork:coverFront	Front cover artwork of the media case containing this music asset
urn:osta-org:mpv:music:artwork:coverBack	Back cover artwork of the media case containing this music asset
urn:osta-org:mpv:music:artwork:leaflet	Front cover artwork of the media case containing this music asset
urn:osta-org:mpv:music:artwork:media	Artwork on the media containing this music asset
urn:osta-org:mpv:music:artwork:artist	Artwork depicting one or more artists of the music.
urn:osta-org:mpv:music:artwork:ensemble	Artwork depicting the performing ensemble, e.g. a band or orchestra.
urn:osta-org:mpv:music:artwork:conductor	
urn:osta-org:mpv:music:artwork:performedBy	
urn:osta-org:mpv:music:artwork:musicBy	
urn:osta-org:mpv:music:artwork:lyricsBy	
urn:osta-org:mpv:music:artwork:recordingLocation	
urn:osta-org:mpv:music:artwork:recordingSession	
urn:osta-org:mpv:music:artwork:performance	
urn:osta-org:mpv:music:artwork:screenCapture	
urn:osta-org:mpv:music:artwork:illustration	
urn:osta-org:mpv:music:artwork:artistLogo	
urn:osta-org:mpv:music:artwork:publisherLogo	
urn:osta-org:mpv:music:artwork:thumbnail	32x32, GIF or PNG

### EXAMPLE

In this example, a JPEG file is related to a MP3 file and identified as an image of a performance of the music asset.

```

...
<mpv:AssetList>
  <!-- This is the per-asset info -->
  <mpv:Audio mpv:id="01-GREAT-SWING-CLASSICS.MP3-20021202031833-a">
    <mpv:LastURL>01 Great Swing Classics.mp3</mpv:LastURL>
    <nmf:Metadata>
      <dc:Properties>
        <dc:creator>Benny Goodman and his Orchestra</dc:creator>
        <dc:format>audio/mpeg</dc:format>
        <dc:title>Jumpin' At The Woodside</dc:title>
      </dc:Properties>
    <mpvm:MusicProperties>

```

```

    <mpvm:AlbumTitle>Great SWING CLASSICS in HI-FI</mpvm:AlbumTitle>
    <mpvm:ArrangedBy>Count Basie;Jimmy Mundy</mpvm:ArrangedBy>
    <mpvm:Genre>Jazz</mpvm:Genre>
    <mpvm:MusicBy>Count Basie</mpvm:MusicBy>
    <mpvm:OrigIndex>1</mpvm:OrigIndex>
    <mpvm:PlayingTime>00:03:28</mpvm:PlayingTime>
    <mpvm:PrincipalArtist>Benny Goodman</mpvm:PrincipalArtist>
    <mpvm:Recorded>1954-11-09</mpvm:Recorded>
  </mpvm:MusicProperties>
</nmf:Metadata>
<mpv:Related mpv:relationship="urn:osta-org:mpv:music:artwork:coverFront">
  <mpv:StillRef mpv:idRef="01-GREAT-SWING-CLASSICS.MP3-20021202031833-R1"/>
</mpv:Related>
<mpv:Related mpv:relationship="urn:osta-org:mpv:music:artwork:performance">
  <mpv:StillRef mpv:idRef="01-GREAT-SWING-CLASSICS.MP3-20021202031833-R2"/>
</mpv:Related>
</mpv:Audio>
<mpv:Still mpv:id="01-GREAT-SWING-CLASSICS.MP3-20021202031833-R1">
  <mpv:LastURL>01 Great Swing Classics.jpg</mpv:LastURL>
  <nmf:Metadata>
    <dc:Properties>
      <dc:format>image/jpeg</dc:format>
      <dc:title>Benny Goodman in concert</dc:title>
    </dc:Properties>
  </nmf:Metadata>
</mpv:Still>
...
</mpv:AssetList>

```

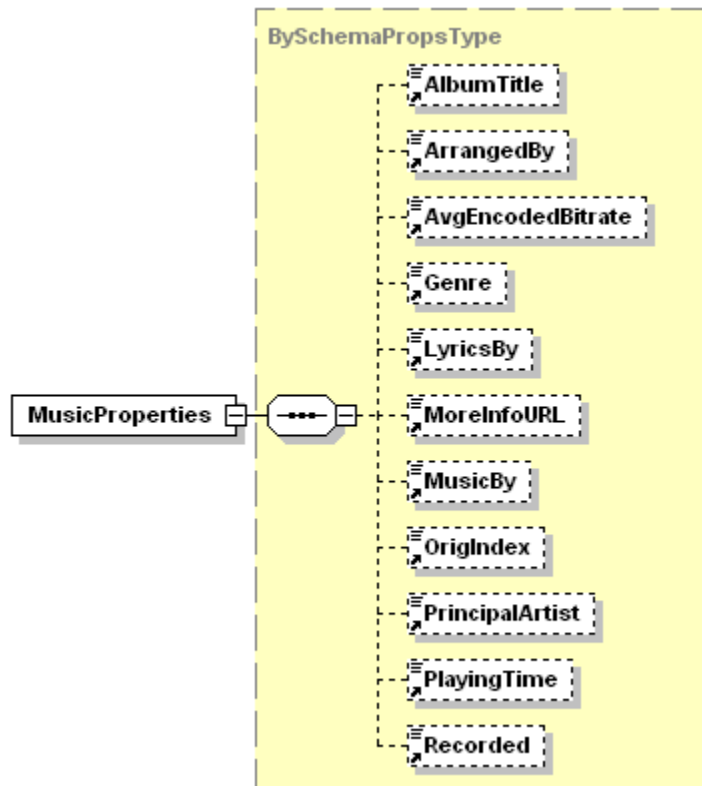
### 5.3 Asset-related Content

The MPV Music Profile allows rich types of artwork for an asset to be specified. A generating application can specify multiple kinds of artwork with any given music asset. The generating application has a choice: it can use the generic “relationship” string of “urn:osta-org:mpv:music:artwork”, or if available, a more specific relationship may be specified.

urn:osta-org:mpv:music:manuscript	The sheet music manuscript. Asset will be of Still, StillMultishotSequence, or Document.
-----------------------------------	--

### 5.4 Music-specific Metadata

#### MPVM:MUSICPROPERTIES



```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://ns.osta.org/mpv/music/1.0/"
xmlns="http://ns.osta.org/mpv/music/1.0/" xmlns:nmf="http://ns.osta.org/nmf/1.0/tools/"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:nmf="http://ns.osta.org/nmf/1.0/"
xmlns:mpvmLyric="http://ns.osta.org/mpv/music/1.0/lyric/" elementFormDefault="qualified"
attributeFormDefault="qualified">
  <xs:import namespace="http://ns.osta.org/nmf/1.0/" schemaLocation="../../imports/nmf/base.xsd"/>
  <xs:annotation>
    <xs:documentation>The MPV Music Properties schema</xs:documentation>
  </xs:annotation>
  <!--
    name for BySchemaProperties element
  -->
  <xs:element name="MusicProperties" type="BySchemaPropsType"
substitutionGroup="nmf:BySchemaPropsBase"/>
  <!--
    top-level schema element type
  -->
  <xs:complexType name="BySchemaPropsType">
    <xs:complexContent>
      <xs:extension base="nmf:BySchemaPropsType">
        <xs:sequence>
          <xs:element ref="AlbumTitle" minOccurs="0"/>
          <xs:element ref="ArrangedBy" minOccurs="0"/>
          <xs:element ref="EncodedBitrate" minOccurs="0"/>
          <xs:element ref="Genre" minOccurs="0"/>
          <xs:element ref="LyricsBy" minOccurs="0"/>
          <xs:element ref="Lyrics" minOccurs="0"/>

```

```

        <xs:element ref="MoreInfoURL" minOccurs="0"/>
        <xs:element ref="MusicBy" minOccurs="0"/>
        <xs:element ref="OrigIndex" minOccurs="0"/>
        <xs:element ref="PrincipalArtist" minOccurs="0"/>
        <xs:element ref="PlayingTime" minOccurs="0"/>
        <xs:element ref="Recorded" minOccurs="0"/>
    </xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:element name="AlbumTitle" type="AlbumTitleType"/>
<xs:complexType name="AlbumTitleType">
    <xs:simpleContent>
        <xs:extension base="xs:string"/>
    </xs:simpleContent>
</xs:complexType>

<xs:element name="ArrangedBy" type="ArrangedByType"/>
<xs:complexType name="ArrangedByType">
    <xs:simpleContent>
        <xs:extension base="xs:string"/>
    </xs:simpleContent>
</xs:complexType>

<xs:element name="EncodedBitrate" type="EncodedBitrateType"/>
<xs:complexType name="EncodedBitrateType">
    <xs:simpleContent>
        <xs:extension base="xs:int"/>
    </xs:simpleContent>
</xs:complexType>

<xs:element name="Genre" type="GenreType"/>
<xs:simpleType name="GenreType">
    <xs:union memberTypes="GenreBaseType xs:anyURI"/>
</xs:simpleType>
<xs:simpleType name="GenreBaseType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Acid"/>
        <xs:enumeration value="Acid Jazz"/>
        <xs:enumeration value="Acid Punk"/>
        <xs:enumeration value="AlternRock"/>
        <xs:enumeration value="Alternative"/>
        <xs:enumeration value="Ambient"/>
        <xs:enumeration value="Bass"/>
        <xs:enumeration value="Blues"/>
        <xs:enumeration value="Cabaret"/>
        <xs:enumeration value="Christian Rap"/>
        <xs:enumeration value="Classic Rock"/>
        <xs:enumeration value="Classical"/>
        <xs:enumeration value="Comedy"/>
        <xs:enumeration value="Country"/>
        <xs:enumeration value="Cult"/>
        <xs:enumeration value="Dance"/>
        <xs:enumeration value="Darkwave"/>
        <xs:enumeration value="Death Metal"/>
    </xs:restriction>
</xs:simpleType>

```



```
<xs:enumeration value="Disco"/>
<xs:enumeration value="Dream"/>
<xs:enumeration value="Electronic"/>
<xs:enumeration value="Ethnic"/>
<xs:enumeration value="Euro-Techno"/>
<xs:enumeration value="Eurodance"/>
<xs:enumeration value="Funk"/>
<xs:enumeration value="Fusion"/>
<xs:enumeration value="Game"/>
<xs:enumeration value="Gangsta"/>
<xs:enumeration value="Gospel"/>
<xs:enumeration value="Gothic"/>
<xs:enumeration value="Grunge"/>
<xs:enumeration value="Hard Rock"/>
<xs:enumeration value="Hip-Hop"/>
<xs:enumeration value="House"/>
<xs:enumeration value="Industrial"/>
<xs:enumeration value="Instrumental"/>
<xs:enumeration value="Instrumental Pop"/>
<xs:enumeration value="Instrumental Rock"/>
<xs:enumeration value="Jazz"/>
<xs:enumeration value="Jazz+Funk"/>
<xs:enumeration value="Jungle"/>
<xs:enumeration value="Lo-Fi"/>
<xs:enumeration value="Meditative"/>
<xs:enumeration value="Metal"/>
<xs:enumeration value="Musical"/>
<xs:enumeration value="Native American"/>
<xs:enumeration value="New Age"/>
<xs:enumeration value="New Wave"/>
<xs:enumeration value="Noise"/>
<xs:enumeration value="Oldies"/>
<xs:enumeration value="Other"/>
<xs:enumeration value="Polka"/>
<xs:enumeration value="Pop"/>
<xs:enumeration value="Pop-Folk"/>
<xs:enumeration value="Pop/Funk"/>
<xs:enumeration value="Pranks"/>
<xs:enumeration value="Psychadelic"/>
<xs:enumeration value="Punk"/>
<xs:enumeration value="R and B"/>
<xs:enumeration value="Rap"/>
<xs:enumeration value="Rave"/>
<xs:enumeration value="Reggae"/>
<xs:enumeration value="Retro"/>
<xs:enumeration value="Rock"/>
<xs:enumeration value="Rock and Roll"/>
<xs:enumeration value="Showtunes"/>
<xs:enumeration value="Ska"/>
<xs:enumeration value="Soul"/>
<xs:enumeration value="Sound Clip"/>
<xs:enumeration value="Soundtrack"/>
<xs:enumeration value="Southern Rock"/>
<xs:enumeration value="Space"/>
<xs:enumeration value="Techno"/>
<xs:enumeration value="Techno-Industrial"/>
```

```

        <xs:enumeration value="Top 40"/>
        <xs:enumeration value="Trailer"/>
        <xs:enumeration value="Trance"/>
        <xs:enumeration value="Tribal"/>
        <xs:enumeration value="Trip-Hop"/>
        <xs:enumeration value="Vocal"/>
    </xs:restriction>
</xs:simpleType>

<xs:element name="LyricsBy" type="LyricsByType"/>
<xs:complexType name="LyricsByType">
    <xs:simpleContent>
        <xs:extension base="xs:string"/>
    </xs:simpleContent>
</xs:complexType>

<xs:element name="Lyrics" type="LyricsType"/>
<xs:complexType name="LyricsType">
    <xs:complexContent>
        <xs:element name="Lang" type="xs:string" minOccurs="0"/>
        <xs:element ref="LyricPart" type="LyricPartType" minOccurs="0" maxOccurs="unbounded"/>
    </xs:complexContent>
</xs:complexType>

<xs:element name="LyricPart" type="LyricPartType"/>
<xs:complexType name="LyricPartType">
    <xs:complexContent>
        <xs:element name="TimeOffset" type="xs:float" minOccurs="0"/>
        <xs:element name="Text" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
    </xs:complexContent>
</xs:complexType>

<xs:element name="MoreInfoURL" type="MoreInfoURLType"/>
<xs:complexType name="MoreInfoURLType">
    <xs:simpleContent>
        <xs:extension base="xs:string"/>
    </xs:simpleContent>
</xs:complexType>

<xs:element name="MusicBy" type="MusicByType"/>
<xs:complexType name="MusicByType">
    <xs:simpleContent>
        <xs:extension base="xs:string"/>
    </xs:simpleContent>
</xs:complexType>

<xs:element name="OrigIndex" type="OrigIndexType"/>
<xs:complexType name="OrigIndexType">
    <xs:simpleContent>
        <xs:extension base="xs:int"/>
    </xs:simpleContent>
</xs:complexType>

<xs:element name="PrincipalArtist" type="PrincipalArtistType"/>
<xs:complexType name="PrincipalArtistType">
    <xs:simpleContent>

```

```

        <xs:extension base="xs:string"/>
    </xs:simpleContent>
</xs:complexType>

<xs:element name="PlayingTime" type="PlayingTimeType"/>
<xs:complexType name="PlayingTimeType">
    <xs:simpleContent>
        <xs:extension base="xs:time"/>
    </xs:simpleContent>
</xs:complexType>

<xs:element name="Recorded" type="RecordedType"/>
<xs:complexType name="RecordedType">
    <xs:simpleContent>
        <xs:extension base="xs:date"/>
    </xs:simpleContent>
</xs:complexType>
</xs:schema>

```

## 5.5 Album/Playlist-level <mpvm:MusicProperties> Music Metadata

The Music Profile defines a schema for music properties. This schema can be used on all audio assets by specifying the root element of the mpvm schema as the only child of the nmf:Metadata element.

The guiding practice for applications and devices that process and present MPV music content based on this schema is that music properties on an mpvp:Album apply also to the tracks contained by that album.

## 5.6 MPV Music Profile Example

```

<?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:mpvm="http://ns.osta.org/mpv/music/1.0/" xmlns:dc="http://ns.osta.org/nmf/1.0/dc/"
  xmlns:nmf="http://ns.osta.org/nmf/1.0/">
  <nmf:Metadata>
    <ManifestProperties xmlns="http://ns.osta.org/manifest/1.0/">
      <ProfileBag>
        <Profile>http://ns.osta.org/mpv/basic/1.0/</Profile>
        <Profile>http://ns.osta.org/mpv/presentation/1.0/</Profile>
        <Profile>http://ns.osta.org/mpv/music/1.0/</Profile>
      </ProfileBag>
    </ManifestProperties>
  </nmf:Metadata>
  <mpvp:Album>
    <nmf:Metadata>
      <dc:Properties>

```

```

    <dc:description>14 swing classics re-recorded in the '50s by th original artists
for great sound with all the integrity and excitement of the original
performances.</dc:description>
    <dc:identifier>7243 5 21223 2 5 Capitol Jazz</dc:identifier>
    <dc:rights>(P) and (C) 1999 Capitol Records, Inc. All rights
reserved.</dc:rights>
    <dc:title/>
  </dc:Properties>
  <mpvm:MusicProperties>
    <mpvm:AlbumTitle>Great SWING CLASSICS in HI-FI</mpvm:AlbumTitle>
    <mpvm:Genre>Jazz</mpvm:Genre>
    <mpvm:MoreInfoURL>www.bluenote.com</mpvm:MoreInfoURL>
  </mpvm:MusicProperties>
</nmf:Metadata>
<mpvp:Foreground>
  <mpv:AudioRef mpv:idRef="01-GREAT-SWING-CLASSICS.WMA-20021202031833-a"/>
  <mpv:AudioRef mpv:idRef="02-GREAT-SWING-CLASSICS.WMA-20021202031833-a"/>
</mpvp:Foreground>
</mpvp:Album>
<mpv:AssetList>
  <mpv:Audio mpv:id="01-GREAT-SWING-CLASSICS.WMA-20021202031833-a">
    <nmf:Metadata>
      <dc:Properties>
        <dc:creator>Benny Goodman and his Orchestra</dc:creator>
        <dc:description/>
        <dc:format>audio/x-ms-wma</dc:format>
        <dc:identifier/>
        <dc:title>Jumpin' At The Woodside</dc:title>
      </dc:Properties>
      <mpvm:MusicProperties>
        <mpvm:AlbumTitle>Great SWING CLASSICS in HI-FI</mpvm:AlbumTitle>
        <mpvm:PrincipalArtist>Benny Goodman</mpvm:PrincipalArtist>
        <mpvm:MusicBy>Count Basie</mpvm:MusicBy>
        <mpvm:ArrangedBy>Count Basie;Jimmy Mundy</mpvm:ArrangedBy>
        <mpvm:AlbumTitle>Great SWING CLASSICS in HI-FI</mpvm:AlbumTitle>
        <mpvm:Recorded>1954-11-09</mpvm:Recorded>
        <mpvm:Genre>Jazz</mpvm:Genre>
        <mpvm:OrigIndex>1</mpvm:OrigIndex>
        <mpvm:PlayingTime>00:03:28</mpvm:PlayingTime>
      </mpvm:MusicProperties>
    </nmf:Metadata>
    <mpv:LastURL>01 Great Swing Classics.wma</mpv:LastURL>
  </mpv:Audio>
  <mpv:Audio mpv:id="02-GREAT-SWING-CLASSICS.WMA-20021202031833-a">
    <nmf:Metadata>
      <dc:Properties>
        <dc:creator>Duke Ellington and his Orchestra</dc:creator>
        <dc:description/>
        <dc:format>audio/x-ms-wma</dc:format>
        <dc:identifier/>
        <dc:title>Harlem Air Shaft</dc:title>
      </dc:Properties>
      <mpvm:MusicProperties>
        <mpvm:AlbumTitle>Great SWING CLASSICS in HI-FI</mpvm:AlbumTitle>
        <mpvm:PrincipalArtist>Duke Ellington</mpvm:PrincipalArtist>
        <mpvm:MusicBy>Duke Ellington</mpvm:MusicBy>
        <mpvm:AlbumTitle>Great SWING CLASSICS in HI-FI</mpvm:AlbumTitle>
        <mpvm:Recorded>1955-11-17</mpvm:Recorded>

```

```
<mpvm:Genre>Jazz</mpvm:Genre>  
<mpvm:OrigIndex>2</mpvm:OrigIndex>  
<mpvm:PlayingTime>00:03:54</mpvm:PlayingTime>  
</mpvm:MusicProperties>  
</nmf:Metadata>  
<mpv:LastURL>02 Great Swing Classics.wma</mpv:LastURL>  
</mpv:Audio>  
</mpv:AssetList>  
</file:Manifest>
```

# Chapter 6: MPV Music Profile Mapping To Other Music Metadata Formats

## 6.1 ID3 and OSTA MPV Music Profile

The ID3 specifications are popular metadata representations for music. The OSTA MPV Music Profile specification provides similar capabilities within the context of the XML-based MPV specification framework.

The following mapping table can be used to associate ID3V1.0 and V1.1 terms and concepts with MPV Music Profile terms and concepts. ID3V2.0 provides much more extensive metadata and is not supported with the MPV Music Profile 1.0.

<i>ID3</i>	<i>MPV Music Profile</i>	<i>Discussion</i>
<b>ID3V1</b>	<b>All specified under mpv:Audio nmf:Metadata</b>	
Song title	dc:title	
Artist	dc:creator	mpvm:MusicProperties:PrincipalArtist can also be used.
Album	mpvm:MusicProperties:AlbumTitle	
Year	mpvm:MusicProperties:Recorded	
Comment	dc:description	
Genre	mpvm:MusicProperties:Genre	
<b>ID3V1.1</b>	<b>All specified under mpv:Audio nmf:Metadata</b>	
Song title	dc:title	
Artist	dc:creator	mpvm:MusicProperties:PrincipalArtist can also be used.
Album	mpvm:MusicProperties:AlbumTitle	
Year	mpvm:MusicProperties:Recorded	
Comment	dc:description	
Album Track	mpvm:MusicProperties:OrigIndex	
Genre	mpvm:MusicProperties:Genre	

## 6.2 WinAMP M3U and OSTA MPV Music Profile

The WinAMP M3U playlist is commonly encountered. The following illustrates mapping M3U playlist to the MPV Music Profile.

<i>M3U Playlist</i>	<i>MPV Music Profile</i>	<i>Discussion</i>
Song title	mpv:Audio nmf:Metadata dc:title	
Filename	mpv:Audio mpv:LastURL	
Duration	mpvm:MusicProperties:PlayingTime	

## 6.3 OSTA MultiAudio and OSTA MPV Music Profile

The OSTA MultiAudio specification provides a CD or DVD table of contents and playlist representation for compressed audio content on data discs. This binary format is suitable for implementation in very resource-constrained devices.

The OSTA MPV Music Profile specification provides similar capabilities within the context of the XML-based MPV specification framework. This allows a single consistent multimedia album format to span music, photo, and video content. For consumer electronics devices able to provide an implementation of the MPV framework, the MPV Music Profile offers a means to support all multimedia content within a constant framework and single firmware implementation.

The following mapping table can be used to associate MultiAudio terms and concepts with MPV Music Profile terms and concepts:

<i>MultiAudio</i>	<i>MPV Music Profile</i>	<i>Discussion</i>
<b>TrackEntry</b>	<b>mpv:Audio asset</b>	
Pathname	mpv:Audio mpv:LastURL	
Track Name	dc:title	
Performer Name	dc:creator	mpvm:MusicProperties:PrincipalArtist can also be used to refine dc:creator.
Composer Name	mpvm:MusicProperties:MusicBy	
Songwriter Name	mpvm:MusicProperties:LyricsBy	
ArrangerName	mpvm:MusicProperties:ArrangedBy	
AlbumName	mpvm:MusicProperties:AlbumTitle	
Genre	mpvm:MusicProperties:Genre	
Playing Time	mpvm:MusicProperties:PlayingTime	
Year Recorded	mpvm:MusicProperties:Recorded	
Track Order	mpvm:MusicProperties:OrigIndex	
Number of Channels		
Average Encoded Bitrate	mpvm:MusicProperties:EncodedBitrate	Use average bitrate for CBR encoded assets.
Maximum Bitrate	mpvm:MusicProperties:EncodedBitrate	Use maximum bitrate for VBR encoded assets.
Sample Rate		
Extra Data	mpv:Metadata and nmf:Metadata	
<b>Playlist</b>	<b>mpvp:Album</b>	

Number of Tracks	-- (implicit)	
Playlist Name	dc:title	
Playlist Description	dc:description	
Track Indexes	mpvp:Album:Foreground contents	
Extra Data	mpv:Metadata and nmf:Metadata	
<b>Playlist Directory</b>	<b>mpvp:Album</b>	
Name	dc:title	
Description	dc:description	
Tracklist Pathnames	mpvp:AlbumRef or mpv:ManifestLinkRef in the mpvp:Album:Foreground	
Playlist Indexes	mpvp:AlbumRef or mpv:ManifestLinkRef in the mpvp:Album:Foreground	
Extra Data	mpv:Metadata and nmf:Metadata	
<b>TOC_Header</b>	<b>file:Manifest nmf:Metadata</b>	
Version Number	encoded into profile and namespace identifiers	
UUID	mpvId:InstanceID	
Volume Name	dc:title	
Data Preparer Identifier		
Publisher Identifier	dc:publisher	
Copyright	dc:rights	
Creation Date and Time	dcterms:created	
Modification Date and Time	dcterms:modified	
Effective Date and Time	dcterms:issued	
Expiration Date and Time		
Number of Playlist Directories	implicit	
Number of Tracks	implicit	
Number of Playlists	implicit	
Extra Data	mpv:Metadata and nmf:Metadata	

...



# Appendix I: 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, 2002.

Available at <http://www.osta.org/mpv/>

**[DCF-1999]**

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

**[DIG35-2001]**

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

<http://www.i3a.org>

**[ISO8601]**

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

**[ISO10646]**

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

**[JFIF]**

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

**[MANIFEST]**

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

**[MD5]**

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

**[MIME-2]**

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

**[MIMETYPES-REG]**

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

**[MPV-Basic]**

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

**[MPV-Core]**

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

**[MPV-Pres]**

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

**[MultiAudio]**

"MultiAudio 1.0"; OSTA, 2001.,  
Available at <http://www.osta.org/>

**[NMF]**

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

**[PNG-MIME]**

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

**[PNG-REC]**

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

**[QT]**

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

**[QT-MIME]**

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

**[RDFsyntax]**

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

**[RDFschema]**

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

**[RFC1766]**

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

**[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/>