

*Digital Music, Photo, and Video Collections*



# **MPV Portable Storage Specification**

**Revision 1.00**

**9 August 2005**

**© 2005 Optical Storage Technology Association**

## POINTS OF CONTACT

<p><u>OSTA</u> David Bunzel OSTA President</p> <p>Tel: +1 (408) 253-3695 d-mail: <a href="mailto:dbunzel@osta.org">dbunzel@osta.org</a></p> <p><a href="http://www.osta.org">http://www.osta.org</a></p> <p><u>MPV Website</u></p> <p><a href="http://www.osta.org/mpv/">http://www.osta.org/mpv/</a></p>	<p><u>Technical Content</u></p> <p>Po-Chieh Hung Chairman, MPV PS Working Group Tel: +81 426-60-9426 e-mail: <a href="mailto:Po-Chieh.Hung@konicaminolta.jp">Po-Chieh.Hung@konicaminolta.jp</a></p> <p>Pieter van Zee MPV Initiative Lead; Editor, MPV Specification Tel: +1 541-715-8658 e-mail: <a href="mailto:Pieter.van.Zee@hp.com">Pieter.van.Zee@hp.com</a></p> <p>Felix Nemirovsky Chairman, MPV Committee Tel: +1 415 643 0944 e-mail: <a href="mailto:felix@chubaconsulting.com">felix@chubaconsulting.com</a></p>
---	--

## ABSTRACT

The Portable Storage specification defines file system structure and naming conventions for long life storage and archiving of digital media content.

## COPYRIGHT NOTICE

Copyright 2005 Optical Storage Technology Association, Inc.

## RELEASE HISTORY

<i>Version</i>	<i>Date</i>	<i>Comments</i>
1.00	08/09/2005	Version 1.0 released.

## ACKNOWLEDGEMENTS

This document is developed by many individuals who dedicate their precious time.

## **IMPORTANT NOTICES (08-AUG-2005)**

- (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 AND SPECIFICATIONS 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) 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.
- (d) 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 AND THE SPECIFICATIONS ARE PROVIDED "AS IS".
- (e) MPV, MusicPhotoVideo AND THE MPV LOGO ARE TRADEMARKS OF OSTA. ALL OTHER TRADEMARKS ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. THE TRADEMARKS MPV, Music/Photo/Video AND THE MPV LOGO MAY NOT BE USED EXCEPT FOR JOURNALISTIC PURPOSES WITHOUT AN EXPLICIT LICENSE FROM OSTA. INFORMATION REGARDING THE MPV TRADEMARK LICENSE IS AVAILABLE AT [www.osta.org/mpv/licensing](http://www.osta.org/mpv/licensing). OBTAINING A LICENSE TO THE OSTA TRADEMARKS MPV, MusicPhotoVideo, AND/OR THE MPV LOGO FROM OSTA DOES NOT PROVIDE A LICENSE TO ANY PATENT REFERENCED HEREIN.
- (f) USE OF OSTA'S MPV TRADEMARKS OR ANY STATEMENT TO THE EFFECT THAT A PRODUCT IS "COMPATIBLE" WITH THE MPV SPECIFICATIONS REQUIRES THE LICENSE DESCRIBED ABOVE. USE OF THE MPV TRADEMARKS WITHOUT THE LICENSE IS PROHIBITED, EXCEPT THAT ATTRIBUTION FOR USE OF THE MPV SPECIFICATIONS MAY BE STATED AS FOLLOWS:  
  
"BASED ON THE MPV™ (MusicPhotoVideo™) SPECIFICATIONS DEVELOPED BY THE OPTICAL STORAGE TECHNOLOGY ASSOCIATION (OSTA). MPV IS A TRADEMARK OF OSTA."
- (g) THE MPV SPECIFICATIONS MAY BE USED ONLY FOR PRODUCTS USED LAWFULLY. OSTA DOES NOT PROMOTE OR CONDONE THE USE OF THE SPECIFICATIONS IN ANY PRODUCT WHICH IS USED FOR AN UNLAWFUL PURPOSE.

# Contents

---

Contents.....	4
Chapter 1: Introduction.....	6
1.1 Executive Summary.....	6
Chapter 2: MPV Portable Storage Specification.....	7
2.1 MPV Portable Storage Introduction.....	7
2.2 Scope.....	8
2.3 Terminology.....	8
2.4 Specification Formal Language.....	9
Chapter 3: MPV Portable Storage File System Structure.....	10
3.1 Summary.....	10
3.1.1 PS Root Directory.....	11
3.1.2 Assets Directory.....	11
3.1.3 Albums Directory.....	13
3.1.4 PS Root Index Manifest File.....	14
3.1.5 Archive Directory (Optional).....	14
3.2 Volume Name.....	15
3.3 PS System Detection.....	15
Chapter 4: MPV Portable Storage – File / Directory Management.....	16
4.1 Basic Procedure.....	16
4.1.1 Copy Asset.....	16
4.1.2 Delete Asset.....	16
4.1.3 Move Asset.....	16
4.1.4 Edit Asset.....	17
4.1.5 Copy Playlist.....	17
4.1.6 Delete Playlist.....	17
4.1.7 Move Playlist.....	17
4.1.8 Edit Playlist.....	17
4.2 Full Maintenance.....	17
4.2.1 Events that Initiate a Maintenance Operation.....	17
4.2.2 Procedure.....	18
4.3 Importing.....	18
4.3.1 From PS System.....	18
4.3.2 From Non-PS system.....	18
4.4 Exporting.....	19
4.5 Remarks.....	19
4.5.1 File and Directory by Alias (or Virtual Link).....	19
4.5.2 File / Directory Protection.....	19
4.5.3 Extra Files and Directories for Application.....	19
4.5.4 Deletion of Asset Roll Directory.....	19
Chapter 5: MPV Portable Storage – Migration.....	20

5.1	Migration by AR Directory Copy .....	20
5.2	Migration by PS Directory Copy .....	21
Chapter 6:	Example of Importing and Exporting .....	23
6.1	DCF/DPOF to PS system.....	23
6.2	PS system to DCF/DPOF.....	23
6.3	Film Scan Disk to PS system.....	24
6.4	Film Scanner to PS system .....	24
Appendix I:	Management of MPV assets with MRD .....	25
I.1	Background .....	25
I.2	MPV Reference Descriptor .....	26
I.2.1	Reference Link and MRD .....	26
I.2.2	Naming Convention for MRD.....	26
I.2.3	Creating and Removing Condition of MRD.....	27
I.3	Basic Operations .....	27
I.3.1	Common Rules.....	27
I.3.2	Create .....	27
I.3.3	Remove.....	28
I.3.4	Update .....	29
I.3.5	Copy .....	29
I.3.6	Move .....	29
I.4	Example.....	29
Appendix II:	References.....	32

# Chapter 1: Introduction

## 1.1 Executive Summary

MPV (MusicPhotoVideo) provides multimedia playlists. MPV 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.

MPV uses a 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.

The MPV specification is being developed in phases and is extended with "profiles". Each profile in MPV defines only those formats and practices that are necessary for the key tasks targeted by the profile. A number profiles are referenced by the MPV Portable Storage Profile, including:

- **Basic Profile:** key tasks: defining content collections, renditions, identifiers, and access to other metadata
- **Presentation Profile:** key tasks: organizing a content collection into a presentation
- **Music Profile:** key tasks: listening to a music collection and interactively browsing content collections

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. The Presentation Profile is also used by the Music Profile 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.

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. OSTA does not warrant that the specification is not or will not be subject to patent claims by other parties.

# Chapter 2: MPV Portable Storage Specification

---

## 2.1 MPV Portable Storage Introduction

One of important functions of digital photography is to preserve current imaging information for the future. In conventional photography using silver halide films, the issue of storage was discussed in the field of materials and preserving environments. In silver halide photography, durability against light, temperature, humidity, physical stress, atmosphere environment are the key elements. However, in digital imaging, in addition to these elements and their effects on storage media such as CD and DVDs, the file system structure should be considered as well. Since digital data can be easily copied, it tends to be dispersed over time. Without a minimum rule for managing files, the image sets may be scattered in the process of iterative updates. As a result, the reliability and ease of finding digital media tends not to be trusted by users. The objective of this standard is to reassure ordinary users that digital picture data is preserved as reliably as silver halide photography.

The DCF (Design rule for Camera File system) standard defines the rules of file / directory naming for removable memory, but is aiming at a handling of a small quantity of image files. Also it does not take into account the merging of two or more DCF file media. A specification is needed that defines the minimum requirements of a directory tree structure which is suitable for storage of thousands of digital still pictures and video for a long period. In order to maintain the structure, such a specification will provide guidelines for the file handling rules such as importing, exporting, modification, and migration procedures to a new medium.

This MPV Portable Storage Profile specification assists users and manufacturers in storage of image files and their management. Unfortunately, the lifetime of digital storage media is not so long as compared to silver halide film because of continuous innovation in media drive readers. Although the media itself could last for a long time, it is likely to be difficult to access the digital media content in the future. To keep user-created content accessible, the data should be migrated to a new medium.

A specification is also needed to define a common guideline to group pictures by "one roll of film", "a common location", "a common event", or "one memory card" or such, since the number of files could include many thousands in a directory. Once the user-created data is stored on a storage medium, a user will have the benefit of electronic file storage - searching and browsing quickly with an easy operation. For instance, the user does not need to flip a page of a paper album - one can see all of pictures on a display.

A specification is needed that defines basic structure of the storage system. Also, guidelines for a common naming convention and handling method for files and directories is desired to reduce the chance of name conflict in

migration as well as to preserve the above mentioned grouping information for the future. It is presumed that a new medium for migration will have more capacity and more flexibility in terms of naming conventions.

## 2.2 Scope

This specification describes the storage method for a large quantity of digital assets (photo, video, and audio), and provides best practices for migrating to a new storage medium, in order to maintain these MPV asset files and their relevant information. This specification is designed especially for ordinary consumers to preserve ordinary digital media content, not for professional nor for specific application-oriented filing system. This specification is typically used for a collection of digital still pictures, video clips and audios generated by users with digital camera using DCF (Design rule for Camera File System) and digital recording in film scanning, etc.

The types of storage media are not limited to existing ones. This specification can be used in any types of storage device such as storage disk, portable media player, digital camera, home server, and personal computer.

## 2.3 Terminology

For the purpose of this MPV Portable Storage (PS) specification, the following definitions apply.

### **album**

Collection of user's asset information such as files and their presentation order.

### **albums directory**

Specified directory that is a child of root that contains manifest files for play-back and templates

### **archive directory**

Specified directory comprises PS archive directories.

### **assets directory**

Specified directory that is a child of root that contains Asset Roll directories.

### **asset list**

Manifest component which identifies assets and handles asset related attributes.

### **asset manifest file**

Particular manifest file for metadata linked to assets. It exists in the AR directory and holds metadata for the assets files.

### **asset roll (AR)**

Group of assets which are typically obtained in a short period. It corresponds to pictures of one roll of film in conventional photography. It may correspond to pictures taken by digital still camera on a removable memory.

### **asset roll directory copy (AR directory copy)**

Copying procedure by the unit of Asset Roll directory.

### **asset roll directory**

Directory contains consumer provided assets, either digitally acquired or scanned from film.

### **index manifest file**

Particular MPV manifest in the root directory file, such as index.pvm, which identifies other manifests.



**migration**

Copying MPV asset files and metadata information from a set of PS media to a new PS medium for inheritance. The medium is written with the same or newer version of the PS specifications.

**MPV asset file**

Data files of photo, video, or audio, which is created by user.

**playlist manifest file**

Particular MPV manifest typically located in the albums directory that describes a particular collection of user assets.

**playlist**

Component in a playlist manifest that describes playback experience.

**Portable Storage directory copy (PS directory copy)**

Copying procedure by a unit of PS directory into the archive directory.

**Portable Storage system (PS system)**

Set of digital music/photo/video files with specified structure of directory trees in this specification.

**PS directory**

Top directory of PS system.

**PS root directory**

Top directory of PS system, which is not under the archive directory.

## **2.4 Specification Formal Language**

The MPV Portable Storage specification uses the following formal language.

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

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

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

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

- Level 1: Error -- **MUST** be fixed.
- Level 2: Severe Warning -- **SHOULD** be fixed to enhance compatibility.
- Level 3: Warning: **OPTIONAL** to be fixed. Not critical to compatibility.

# Chapter 3: MPV Portable Storage File System Structure

## 3.1 Summary

Portable Storage has a specified file system structure for the storage of MPV asset files with a unified manner based on the concept of unique ID. The PS root directory consists of the assets directory, the albums directory, and the index manifest file as mandatory, and archive directory as optional. The assets directory consists of any number of asset roll directories, and the albums directory consists of any number of playlist manifest files. The archive directory consists of other PS directories, which may be migrated from another Portable Storage system by PS directory copy. The assets directory, the albums directory and an index manifest shall be prepared in the PS root directory even if they are empty.

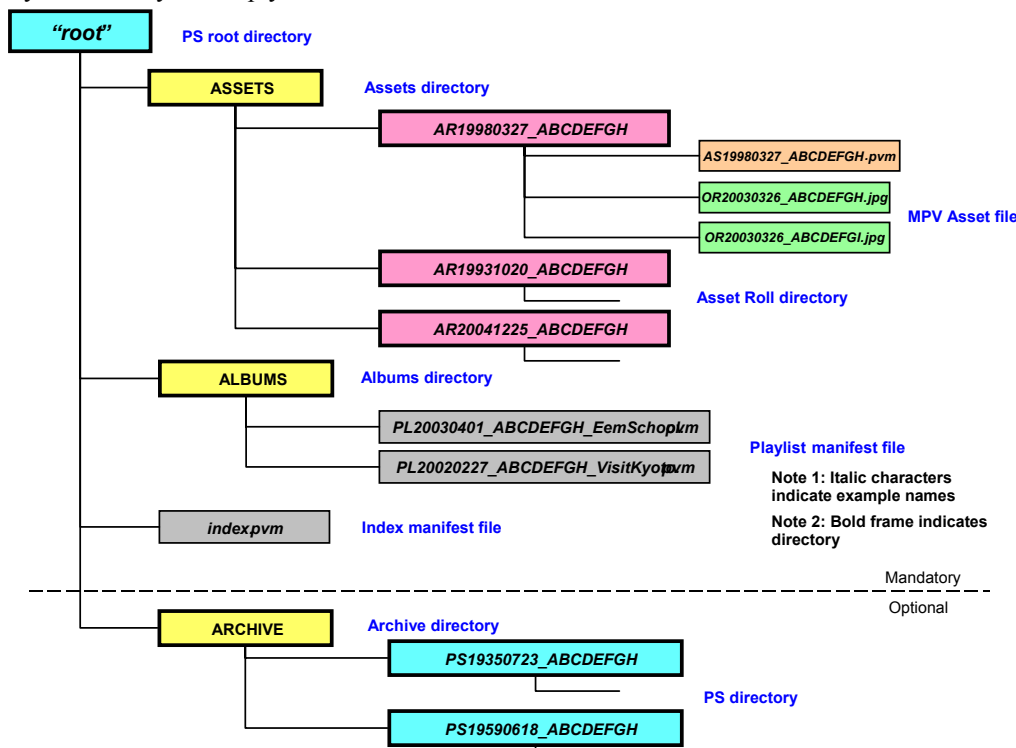


Figure 1. Illustration of general structure of the Portable Storage system

The PS root directory can hold any user specific data such as executable code for slide shows on a PC and data files for a high-speed search, etc. However, these are regarded as non-PS compliant objects, and may not be migrated to a new medium.

### **3.1.1 PS Root Directory**

The PS root directory can have any name or location in the filesystem. It is preferred to be the root directory of a volume. If the directory is not the medium root, the best practice is to use the naming rule described in Section 3.1.5.1.

### **3.1.2 Assets Directory**

The assets directory is a mandatory sub-directory under PS (root) directory. It stores any numbers of asset roll directories. As same as PS root directory, any data and files are allowed to be stored in this directory as long as reserved names are not used. However, they are not guaranteed to be migrated.

The name of assets directory shall be:

"ASSETS" (capital)

#### **3.1.2.1 Asset Roll Directory**

Asset roll directory is a sub-directory located under the assets directory. Dozens of MPV asset files that are taken in a short period and/or in a local area should be stored in an asset roll directory with an asset manifest file.

The number of images in an asset roll directory (here, the number of images is counted as one when several resolutions exist in an asset roll directory) is suggested to be less than one hundred in cases where there is no other natural size for the number of images. Note that this maximum number is derived from the maximum number that an ordinary user can view at once and recognize the contents on a display or a sheet of print.

A set of MPV asset files should be stored into an AR directory based on the following guidelines:

- (1) When the original MPV asset files are scanned from one roll of film,
- (2) When the original MPV asset files stored in a directory in another storage system, and
- (3) When the original MPV asset files stored in one medium.

Any files can be stored in this directory. The following naming convention is applied for asset roll directory:

##### ***3.1.2.1.1 New name***

The name of asset roll directory must consist of date and unique code. The directory name starts with "AR" (capital), and is followed by year, month, date, underscore, unique code and optional characters.

"ARyyymmdd\_uuuuuuuu\_pppppp"

where,

yyyy: year in 4 digits

mm: month in 2 digits

dd: day in 2 digits

\_uuuuuuuu: 8-digit unique code (36-type ASCII: 0 - 9, A-Z)

\_pppppp: Optional characters, start with "\_" (underscore) if there are any. The character encoding is UTF-8.

There is no restriction for the number of characters as long as the total byte length is equal to or less than 255.

The 8-digit unique code is generated by the seed number derived from current time information, such as "hhmmss.cc". If more than one directory is supposed to be generated with the same day information, the late-generated directory has a bigger value in the unique code so as to be sorted by the name. The 8-digit unique code may be generated from the time information and other unique code such as MAC (Media Access Control) address.

Note: UUID/GUID requires 128-bit code but since the name already has date information, a smaller number of digit is suggested so as to compromise between complexity of name and uniqueness. Since there is a rare chances to see a conflict in copying or migration procedure, there is a renewal rule to be described below.

The date information can be assigned from one of the following guidelines:

- (1) Choose a date from the first day to the last day when the MPV asset files in the directory were generated, if the date information is available. The date may be estimated by metadata of image files when they are captured by digital still camera. When they are digitized by scanner from a silver-halide film or print, the date could be estimated from date information exposed onto film, film type, date information recorded by APS, etc. Note that the create date and the capture date in the file may not be accurate.
- (2) If these dates are unknown, or are believed to be unreliable, the date of digitization should be used instead.
- (3) If the date information is not exactly known, "00" (zero-zero) should be used for day and/or month. The best estimate should be used for year information.

### ***3.1.2.1.2 Convention for resolving naming conflicts***

In case that the renewal of a directory name is necessary due to a conflict in copying, migration, or modification, the unique code is incremented until no conflict is found. The conflict is judged without optional characters.

(original) ARyyyyymmdd\_uuuuuuuu  
(regenerated) ARyyyyymmdd\_vvvvvvvv  
vvvvvvvv: Add one to the original "uuuuuuuu"  
Example: "D0DE9RCT" will be "D0DE9RCU".

Note: If the unique code "vvvvvvvv" reaches the upper limit ("ZZZZZZZZ"), it starts from "00000000".

## **3.1.2.2 MPV Asset Files**

The types of file format allowed in the portable storage system should be chosen from published international and/or national standards.

If thumbnail and/or screen resolution images are necessary in the same directory, the same name with different extension name may be used as defined in DCF, or new sub-directory in the AR directory may be used.

### ***3.1.2.2.1 Naming of MPV asset file***

The name of an MPV asset file must be given in the same fashion. The file name starts with "OR" (capital. It stands for ORiginal), and is followed by year, month, date, underscore, and unique code.

ORyyyyymmdd\_uuuuuuuu\_pppppp.ext:  
yyyy: year in 4 digits  
mm: month in 2 digits  
dd: day in 2 digits  
\_uuuuuuuu: 8-digit unique code  
\_pppppp: Optional characters (Start with "\_" (underscore) if there is. The character encoding is UTF-8.  
There is no restriction for the number of characters as long as the total byte length is equal to or less than 255, including the extension.)  
ext: Extension name indicates the format of file

The unique code may be identical to the one used in the AR directory. If more than one file is stored with the same day information, the unique code should be incremented in order. If it reaches to the upper limit (i.e., "ZZZZZZZZ"), the starting unique code should be chosen again, not to reach the upper limit.

### ***3.1.2.2 Convention for resolving naming conflicts***

In case of the renewal of the MPV asset file name due to editing, increment the unique code until no conflict is found. Conflict should be judged without optional characters.

(original) ORyyyymmdd\_uuuuuuuu  
(regenerated) ORyyyymmdd\_vvvvvvvv  
vvvvvvvv: Add one to the original "uuuuuuuu"

If "uuuuuuuu" reaches the limit ("ZZZZZZZZ"), it starts from "00000000" until no conflict is found.

### **3.1.2.3 Asset Manifest File**

An asset manifest file consisting of the list of assets in the AR directory is mandatory and must be stored with the specific name. The format must be compatible with the MPV specifications. The name of asset manifest file must start with "AS" (it stands for "ASset" as follows:

ASyyyymmdd\_uuuuuuuu.pvm  
yyyymmdd\_uuuuuuuu: The identical code with the directory name

The file must include information given for the MPV asset files in the directory.

### **3.1.3 Albums Directory**

The albums directory is a mandatory sub-directory under PS (root) directory. The name of albums directory shall be:

"ALBUMS" (capital)

#### **3.1.3.1 Playlist Manifest File**

Playlist manifest file is used to specify a list of MPV asset files in the portable storage, mainly for a playback of preferred assets with appropriate transient effect from one image to another. The file should use the information of relative paths to images from the location of the playlist manifest file. The playlist manifest file must be written according to the MPV specifications. A playlist manifest file corresponds to a photo / music album. The manifest file should include information given for the album.

In the case the user creates a custom playlist, the name of a playlist manifest file must have the prefix "PL" (it stands for PPlaylist) followed by the generation date of the playlist manifest file, the unique code described 3.1.2.1.1 and optional characters in order to avoid a conflict in migration. For example, a playlist manifest file generated on July 22, 2004 may be:

"PL20040722\_uuuuuuuu\_TripToCalifornia.pvm"

In the case that the playlist file is a presentation of all of the assets that are stored in an AR directory, the same characters for the AR directory must follow the prefix "PL." For instance, the playlist for an AR directory "AR20040722\_uuuuuuuu" will be:

"PLAR20040722\_uuuuuuuu.pvm"

### **3.1.4 PS Root Index Manifest File**

The PS root directory must have an index manifest file that will be found according the MPV Core specification naming rules. The file should be named index.pvm. The index file must reference all the playlist manifest files. This allows a simple MPV browser device or application to explore the contents of a PS system medium simply by navigating from the index manifest file.

### **3.1.5 Archive Directory (Optional)**

The archive directory is the optional sub-directory under the PS root directory, which comprises PS directories that may have been already completed in another medium, and are not supposed to be modified anymore. A specific suggested use is when an existing directory structure is to be imported without restructuring it to conform to the requirements of the assets directory.

The name of PS archive directory shall be;

"ARCHIVE" (capital)

#### **3.1.5.1 PS Directory Name in PS Archive Directory**

Unlike the PS root directory, PS directory in the archive directory is recommended to have specified name according to the following rule.

##### ***3.1.5.1.1 New name***

The name of PS directory consists of date and unique code. The date is determined from either;

- (1) When the PS directory is created, or
- (2) The last day of AR directories inside the PS directory

The directory name starts with "PS" (capital), and is followed by year, month, date, underscore, unique code, and optional characters.

PSyyyyymmdd\_uuuuuuuu\_ppppppp

yyyy: year in 4 digits

mm: month in 2 digits

dd: day in 2 digits

\_uuuuuuuu: unique code

\_pppppp: Optional characters (Start with "\_" (underscore) if there is. There is no restriction for the number of characters as long as the total character is equal to or less than 255.)

##### ***3.1.5.1.2 Convention for resolving naming conflicts***

In case of the renewal of PS directory name, the unique code should be incremented until no conflict is found. A conflict should be judged without the optional characters.

(original) PSyyyyymmdd\_uuuuuuuu

(regenerated) PSyyyyymmdd\_vvvvvvvv

vvvvvvvv: Add one to original "uuuuuuuu"

If unique code "vvvvvvvv" reaches to the limit ("ZZZZZZZZ"), it starts from "00000000" and is incremented until no conflict is found.

## 3.2 Volume Name

In many cases, the PS system will be constructed in a medium such as CD or DVD. Both a media specific name and a PS-specific name may be used. The following naming rule is suggested.

The date of volume creation should be used. It starts with year, month, day, 8-bit unique code and optional characters.

yymmdduuuuuuu\_ppppppp

yy: year in 2 digits

mm: month in 2 digits

dd: day in 2 digits

uuuuuuu: 8-bit unique code. The code may be identical to the unique code of the oldest AR directory created (See 3.1.2.1.1). If the length of volume name is limited, use the characters from the start for as many as possible.

Some storage media may not allow the full length name to be used; in such case, it should be shortened using an implementation-specific algorithm.

In addition, the index manifest file should store a full ID.

## 3.3 PS System Detection

A storage medium may be scanned to detect one or more MPV PS systems. The following file system structures must be detected:

- The PS root may be at an arbitrary location in the host storage media file system
  - root
- Presence of a root index manifest
  - Such as root/INDEX.PVM
- Presence of the assets directory
  - root/ASSETS
- Presence of at least one asset roll directory
  - root/ASSETS/ARyyyymmdd\_uuuuuuuu\_pppppp
- Presence of at least one asset roll manifest file
  - root/ASSETS/ARyyyymmdd\_uuuuuuuu\_pppppp/ASyyyymmdd\_uuuuuuuu.pvm
- Presence of at least one original file
  - root/ASSETS/ARyyyymmdd\_uuuuuuuu\_pppppp/ORyyyymmdd\_uuuuuuuu\_pppppp.ext
- Presence of the albums directory
  - root/ALBUMS

# Chapter 4: MPV Portable Storage – File / Directory Management

---

In order to preserve the directory tree structure of Portable Storage, it is suggested to handle MPV asset files and directories as follows.

## **4.1 Basic Procedure**

### **4.1.1 Copy Asset**

When a new file is copied into an AR directory, the following procedure should be performed:

- (1) Check the filename.
- (2) If the file name is not compliant with the recommendation in this specification, a new name should be given based on the rule mentioned above.
- (3) If there is no conflict, simply copy it. If there is a conflict, rename it based on the rule mentioned above.
- (4) Add the reference to the new file in the asset manifest file in the AR directory.

### **4.1.2 Delete Asset**

The following procedure should be performed:

- (1) Delete the file
- (2) Find references to the old name, replace with the new name, in the asset manifest file in the AR directory.
- (3) Search playlist manifest files in the albums directory.
- (4) Find references to the old name, delete in all playlist manifest file(s) if applicable.

Deleting procedure should be minimized because the searching playlist manifest files having the reference link to the MPV asset file may take a long time.

### **4.1.3 Move Asset**

Use the copy and delete procedure for the MPV asset files for the original directory and destination directory, respectively. In addition:

- (1) Search playlist manifest files in the albums directory.
- (2) Find references to the old name, replace with the new name, in all playlist manifest file(s) if applicable.



#### **4.1.4 Edit Asset**

If the original file is modified, it is recommended that regenerated filename be given by adding or modifying the optional characters, rather than overwrite it.

#### **4.1.5 Copy Playlist**

When a playlist manifest file is copied into the albums directory, the following procedure should be performed:

- (1) Check the filename.
- (2) If the file name is not compliant with the requirements in this specification, a new name should be given based on the rule mentioned above.
- (3) If there is no conflict, simply copy it. If there is a conflict, rename it based on the rule mentioned above.
- (4) Find references to the asset files in the playlist and use the copy asset procedure for all assets not in the PS system.
- (5) Add the playlist file to the index manifest file.

#### **4.1.6 Delete Playlist**

The following procedure should be performed to delete a playlist manifest file:

- (1) Delete the file
- (2) Search playlist manifest files in the albums directory.
- (3) Find references to the playlist manifest name, delete in all playlist manifest file(s) if applicable.
- (4) Search the index manifest file in the PS root directory.
- (5) Find references to the playlist manifest name, delete in the index manifest file.

Deleting procedure should be minimized because the searching playlist manifest files having the reference link to another playlist file may take a long time.

#### **4.1.7 Move Playlist**

Use the copy and delete procedure for the MPV playlist manifest files for the original playlist and destination playlist, respectively. In addition:

- (2) Search playlist manifest files in the albums directory.
- (3) Find references to the playlist manifest name, delete in all playlist manifest file(s) if applicable.
- (4) Search the index manifest file in the PS root directory.
- (5) Find references to the playlist manifest name, change in place in the index manifest file.

#### **4.1.8 Edit Playlist**

If the original file is modified, it is recommended that the user may choose whether to overwrite it or create a new copy.

## ***4.2 Full Maintenance***

Since Portable Storage is an open system, there are chances to be modified by non-PS system such as a personal computer if the medium is rewritable. As the result, the relationship between file system structure and index files might be destroyed.

In order to fix the relationship, a PS compliant system should have a capability to perform the full maintenance operation from time to time.

### **4.2.1 Events that Initiate a Maintenance Operation**

The full maintenance operation should be performed when at least one of the following conditions is matched.

- (1) When migration occurs, or
- (2) When user requires.

### **4.2.2 Procedure**

In the full maintenance, the following maintenance should be performed.

- (1) Check that asset manifest file has descriptions for all of the MPV asset files in AR directories.
- (2) Check that playlist manifest file(s) has corresponding asset manifest files and MPV asset files.

In case (1), if a corresponding asset description in the manifest file is missing, the appropriate asset description should be added.

In cases (1) and (2), if a corresponding MPV asset file is missing, the asset description in the manifest file may be deleted or the corresponding asset may be searched.

## **4.3 Importing**

### **4.3.1 From PS System**

There are two recommended copying procedures: namely, AR directory copy and PS directory copy.

#### **4.3.1.1 AR Directory Copy**

An AR directory can be copied as it is into the assets directory if the original file system is compliant with the PS specification. In case that the directory name conflicts with another asset roll directory of the counterpart, try to identify the both AR directories have the same contents. If these are identical, it is recommended to inquire about the decision for user whether it would be overwritten or copy with a regenerated name. See 5.1.

#### **4.3.1.2 PS Directory Copy**

A PS directory is copied into the Archive directory. In case that the directory name conflicts with PS directories of the counterpart, the same procedure described 3.1.5.1.2 should be applied. Otherwise, it should be copied without any modifications. See 5.2.

### **4.3.2 From Non-PS system**

The following is a general approach for the importing operation for non-PS system. For the importing from the DCF system, see Chapter 6 for more detail.

#### **4.3.2.1 When MPV Asset Files are Already Stored in a Directory**

The original directory should be copied to the destination PS assets directory with a new AR directory name and new MPV asset file names.

#### **4.3.2.2 When MPV Assets Files are stored without organization**

Assets should be re-grouped by a certain category such as create time or location, and should be copied into a newly created AR directory at the destination PS system. If grouping is not possible, all of the assets can be copied into a newly created AR directory.

## **4.4 Exporting**

No general approach is specified for exporting. For the exporting to the DCF system, see Chapter 6 for more detail.

## **4.5 Remarks**

### **4.5.1 File and Directory by Alias (or Virtual Link)**

It is not specified in this specification. There may be a chance not to be migrated to a new medium in the future since it is depending on operating system.

### **4.5.2 File / Directory Protection**

Follow the specification of medium or operating system.

### **4.5.3 Extra Files and Directories for Application**

Allowed. These may have a chance not to be migrated to a new medium in the migration process.

### **4.5.4 Deletion of Asset Roll Directory**

Deletion of asset roll directory should be minimized. When an asset roll directory has to be deleted by some reasons, playlist manifest files and the index manifest file should be maintained accordingly.

# Chapter 5: MPV Portable Storage – Migration

---

When a new medium having more capacity is available, the migration operation should be considered. Migration occurs when two or more PS systems will be merged into a unified PS system. There are two types of guidelines for migration as follows.

## ***5.1 Migration by AR Directory Copy***

Each asset roll is copied into the PS assets directory of the destination medium. Also, playlist manifest files are copied into the PS albums directory, and should be maintained. The new index manifest file, playlist manifest file and asset list manifest file should be maintained to secure reference links and to avoid conflicts in IDs.

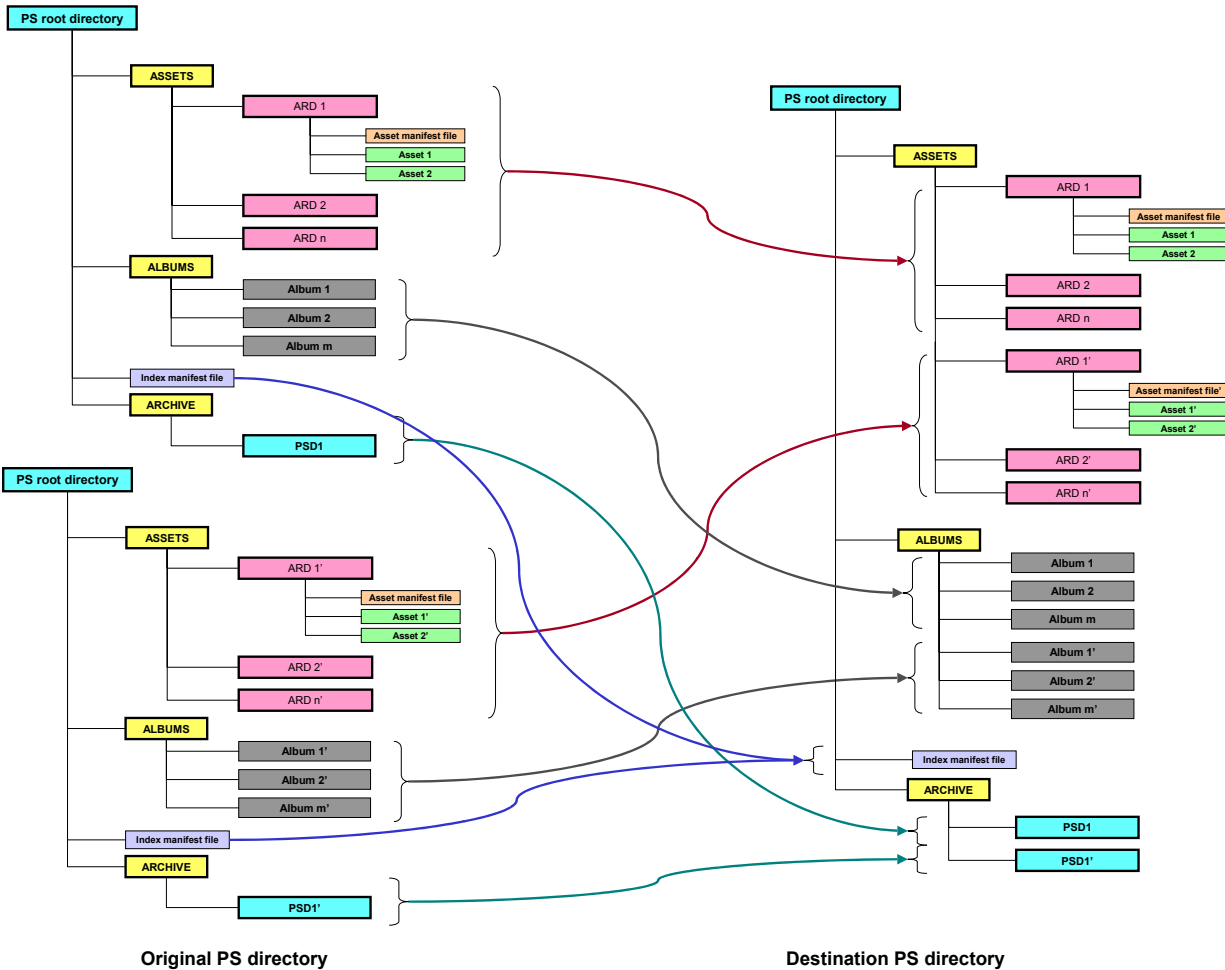


Figure 2 Migration by AR directory copy

## 5.2 Migration by PS Directory Copy

The PS root directory is copied into the archive directory of the destination PS system, which is typically the latest PS system. If the destination PS system already has the archive directory, the original PS directory should be placed into the PS archive directory. The only index manifest file in the PS root directory should be maintained to have a link to the index manifest file in PS directory under the archive directory.

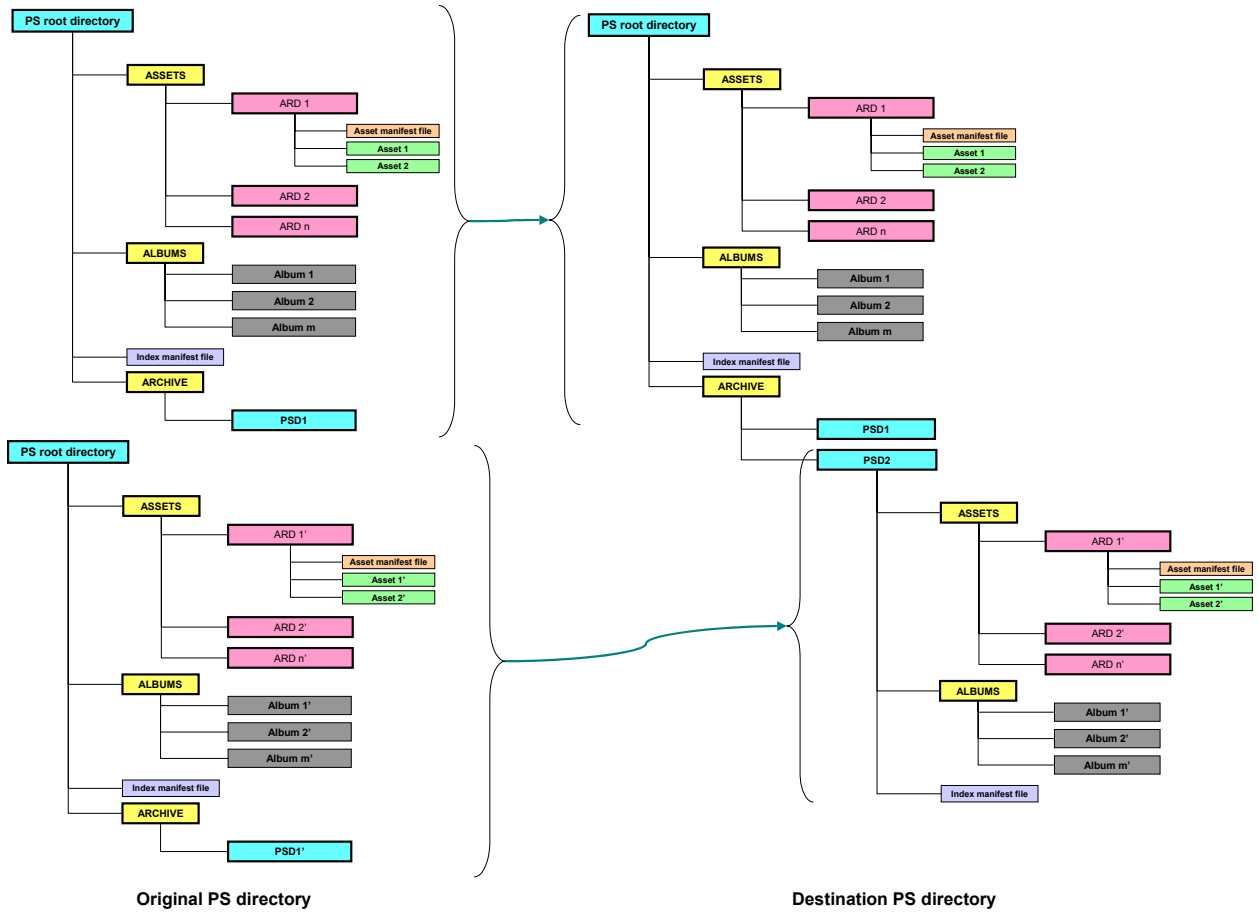


Figure 3 Migration by PS directory copy

# Chapter 6: Example of Importing and Exporting

---

## 6.1 DCF/DPOF to PS system

In copying from DCF/DPOF to PS system:

- (1) Change directory name; i.e. "DCIM" to "ASSETS"
- (2) Change directory names; e.g. "123CMPNY" to "ARyyyymmdd\_uuuuuuuu"
- (3) Change file names, e.g. "PICTssss.jpg" to "ORyyyymmdd\_uuuuuuuu.jpg"
- (4) Create an asset manifest file in the AR directory.
- (5) Change directory name; i.e. "MISC" to "ALBUMS," and convert DPOF files to playlist manifest files if necessary.

Note: When two DCF objects share the same name but have different extensions on the DCF file system (such as .jpg and .wav), the same name is allowed to be given to both files in the PS system as long as the extensions are different.

## 6.2 PS system to DCF/DPOF

In copying from PS system to DCF/DPOF:

- (1) Change the PS assets directory name; e.g. "ASSETS" to "DCIM"
- (2) Change AR directory names; e.g. "ARyyyymmdd\_uuuuuuuu" to "100PSARD" (PSARD stands for PS Asset Roll Directory.)
- (3) Change file names; e.g. "ORyyyymmdd\_uuuuuuuu.jpg" to "PICTssss.jpg"

Note: Directories and files which are not defined in DCF may be ignored. The serial number "ssss" will be created in lexicographic order, based on date information and the unique ID code.

Table 6-1 Comparison of DCF and Portable Storage

DCF name	Example	name in PS	Example
/	/	PS root directory	/
DCF Image Root directory	DCIM	assets directory	ASSETS
DCF directory	100CMPNY	asset roll directory	AR20040722_ABCDEFGH
DCF object	PICT0001.jpg	MPV asset file	OR20040722_ABCDEFGH.jpg
	PICT0002.jpg		OR20040722_ABCDEFGI.jpg

### 6.3 Film Scan Disk to PS system

The following procedure will be applied.

- (1) Since one roll of image files is already stored in one directory, the directory is copied to a new AR directory. e.g. "Pictures" to "ARyyymmdd\_uuuuuuuu"
- (2) Request user to input the correct date of shooting photography into the asset manifest file. If it is impossible, the date of digitization is used.
- (3) Change file names, e.g. "xxxxxx.jpg" to "ORyyymmdd\_uuuuuuuu.jpg"

Note: Examples of film scan disk are "Picture CD," "Fujicolor CD," etc.

### 6.4 Film Scanner to PS system

The following procedure will be applied.

- (1) Decide one of following rules to determine the directory name.
  - Date given by photographer
  - Date estimated from the contents of the photography
  - Date estimated from film type
  - Scanning date
- (2) Generate MPV asset filenames as described in this specification.



# Appendix I: Management of MPV assets with MRD

---

This appendix is informative (optional) in this version of the PS specification. It is prepared to note a potential problem that might occur in the future and to propose a way to resolve it. In this version, the problem is supposed to be solved in the full maintenance operation described in 4.2.

## I.1 Background

From the nature of MPV, the basic handling unit of Portable Storage is (playlist or asset) manifest file. It needs more effort to maintain the representation consistency between MPV asset files or directories and the metadata in the MPV manifests. The manifest file is defined by XML description in MPV Manifest specification and it refers to at least one or more MPV asset files that are the basic handling unit of the real file system. The descriptions on relations between manifest file and the referenced real MPV asset files are able to be distributed over storage. The relation can vary from 1:1 to n:n, which can be newly created relation or changed one. These characteristics lead to a problem namely “broken-link” problem, which means the metadata in one or more MPV manifests does not match to the real MPV asset file or is missing in the storage. It occurs when an MPV asset file is deleted, renamed or moved. It is critical when a certain MPV asset file is cross-referenced which often occurs in the MPV system. Another problem is “metadata synchronization” which occurs when the metadata of a certain asset is duplicated and distributed into a set of manifest. If one of the metadata is updated, it is difficult for a small system to update the other metadata in real time.

In practices, managing the MPV asset files and the metadata and keeping the consistency for fixed storage systems is an implementation issue. It can be achieved by other technologies such as database and special file system and so on, in addition to the full maintenance operation. However, these processes may take a long time for system with poor processing power.

For PS system, one of the typical storage media is a removal type, which requires common handling method among device vendors. It is desired not to have vendor specific system to preserve the directory tree structure or to keep the consistency among MPV manifests and MPV asset files. Thus, in this appendix, it is exemplified that the PS system maintains the special information namely “backward reference link” and operates using it to resolve the problems efficiently.

## I.2 MPV Reference Descriptor

The MPV technology is based on XML which describes references in one-way, from parent to child. When some operations such as “add,” “remove,” “rename,” “move,” and “copy” occur, the one-way reference link does not provide any information of the depending asset or manifest link.

When a certain MPV asset file is cross-referenced by multiple asset descriptions in MPV manifests, a “Reference Link” is supposed to be stored in a specific file called “MRD (MPV Reference Descriptor),” which describes the dependency information.

### I.2.1 Reference Link and MRD

The reference link describes which manifest refers to the MPV asset file. It is described in XML and the grammar is much similar to the MPV’s. The namespace prefix is “ps:” and there are only two schema elements: “ReferenceLink” and “LastURL”

An MRD is a file that stores multiple reference links. Its name is specified by the MRD naming convention described below. The existence of MRD should be determined by the reference count to the MPV asset file.

#### I.2.1.1 <ps:ReferenceLink> -- a Set of Reference Link Identifier

A PS ReferenceLink is an unordered set of reference link locations. It is a unique identifier in an MRD.

```
<ps:ReferenceLink xmlns:ps="http://ns.osta.org/ps/1.0/"> ... </ps:ReferenceLink>
```

#### I.2.1.2 <ps>LastURL> -- Reference Link Location Identifier

A PS LastURL is a path to last known location and the mpv\_id of referenced asset description. The syntax of PS LastURL follows the URI syntax like <mpv>LastURL> which is described in the MPV Core. It is recommended to describe the path in relative manner as follows:

```
<ps>LastURL>../../index.pvm?mpv_id=cover0</ps>LastURL>
```

#### I.2.1.3 Typical MRD

A typical example of MRD is as follows. In this case, an MPV asset file, “ASSETS/AR19980302\_ABCDEFGH/OR20030326\_ABCDEFGH.jpg” is referenced by three times from manifest files.

```
<!-- asset/AR19980302 ABCDEFGH/OR20030326 ABCDEFGH.mrd -->
<?xml version="1.0"?>
<ps:ReferenceLink xmlns:ps="http://ns.osta.org/ps/1.0/">
  <ps>LastURL>AS19980327 ABCDEFGH.pvm?mpv_id=photo00000001</ps>LastURL>
  <ps>LastURL>../../album/PL20030401 ABCDEFGH ElemSchool.pvm?mpv_id=portrait0</ps>LastURL>
  <ps>LastURL>../../album/PL20040713_01234567_MyPhotos.pvm?mpv_id=bigsize</ps>LastURL>
</ps:ReferenceLink>
```

### I.2.2 Naming Convention for MRD

An MRD filename should be assigned as follows:

- (1) If a manifest file refers to only one MPV asset file (simple asset), the filename of the MRD is the cocatenation of the original MPV asset file, an underscore and its extension with replacement of the extension to “.mrd”. If the combined name exceeds 255 bytes, it must be truncated from the end of the original basenane until the size is 255.

- For example, if an MPV asset filename was “OR20041129\_RS0T29IM\_cover.JPG” and it is the only MPV asset file, which is referenced by the manifest file, the MRD filename would be determined to be “OR20041129\_RS0T29IMrs0t29im\_cover.JPG.mrd”.
- (2) If a manifest file refers indirectly to more than one MPV asset file with simple assets (composite asset), the filename of the MRD is same as the first MPV asset file, which is referenced by the first simple asset, from the beginning of simple asset list in the composite asset.

### **I.2.3 Creating and Removing Condition of MRD**

An MRD should be only created in the following conditions:

- (1) When the cross-reference link has occurred in an MPV manifest. In other words, when a new asset description is created based on an existing asset description within the MPV manifest.
- (2) When the cross-reference link has occurred in multiple MPV manifests. In other words, when a new asset description is created in an MPV manifest based on an existing asset description in the other MPV manifests.

The MRD should be removed at the following conditions:

- (1) When the MPV asset file that has the same filename as the MRD file is removed.
- (2) When only one reference link remains within the MRD file.

As a result, the absence of MRD means the asset description is not cross-referenced but just is one-to-one referenced.

## **I.3 Basic Operations**

To keep the consistency among MPV asset files and MPV asset descriptions, PS system with MRD should follow procedures as below.

### **I.3.1 Common Rules**

The followings are common rules for the all operations:

- (1) Each operation should be performed in a single unit.
- (2) Each operation should be performed in a simple asset unit. For the composite asset, the operations are performed by a combination of simple asset units.

By the nature of MPV, none of MPV asset files in the PS system is supposed to exist without any reference from a manifest file. The operations by user of the PS system should always be performed through an MPV parser.

### **I.3.2 Create**

There are two cases in the create operation in the PS system. In the case that a new MPV asset file is created in an AR directory, a new simple MPV manifest file that references the MPV asset file should be created. If the MPV asset file is newly generated, the filename should be assigned according to the PS MPV asset file naming convention. If the MPV asset file is just copied from outside of the AR directory or from non-PS system, the following procedure should be performed to create the MPV asset file:

- (1) Check the name of the file that is copied from outside of the AR directory, whether the filename is compliant with the PS naming convention or not.
- (2) If the filename is not compliant, a new name should be assigned according to the PS MPV asset file naming convention.
- (3) If the filename is compliant or newly assigned by the second step, copy the MPV asset file. If it is not a unique filename, rename it based on the PS MPV asset file naming convention.

(4) Add the new simple MPV asset description that refers to the MPV asset file into the asset manifest file in the AR directory.

Note: In this case, it is not necessary to create or update MRD.

Creating a new simple MPV asset description in an MPV manifest does not always mean creating a new MPV asset file. It could be created by referencing an existing MPV asset file while browsing an existing MPV manifest. In this case, it is not necessary to create or to rename the MPV asset file, but just use it as it is. The following procedure should be performed to create a new simple MPV asset description that references it to keep the consistency:

- (1) Add the new simple MPV asset description that refers the existing MPV asset file into the MPV manifest (e.g., album).
- (2) Check whether the corresponding MRD exists or not. The corresponding MRD can be determined by filename.
- (3) If the MRD file does not exist, create it with a “set of reference link” (<ps:ReferenceLink>) with “reference link” (<ps:LastURL>), which point to the original simple MPV asset description in the existing MPV manifest(s).
- (4) Add the reference link that points the newly created simple MPV asset description in the MRD, by appending the reference link.

Creating a composite MPV asset description in an MPV manifest means that there are a set of simple asset descriptions that point to the existing MPV asset files. In this case, the following procedure should be performed to create a new composite MPV asset description:

- (1) Add the new composite MPV asset description that refers the existing simple asset descriptions into the MPV manifest.
- (2) For one of the simple asset descriptions which the composite MPV asset point to, check whether the corresponding MRD exists or not. This is determined by filename.
- (3) If the MRD file does not exist, create it with a “set of reference link (<ps:ReferenceLink>)” and “a reference link (<ps:LastURL>)”, which points to the simple asset.
- (4) Add the reference link that points to the newly created composite MPV asset, by appending the set of reference links.
- (5) Iterate steps from (2) to (4) for each simple asset description that is pointed by the newly created composite MPV asset.

### **I.3.3 Remove**

While browsing through a simple MPV asset description in an MPV manifest, there are two cases to remove a simple asset description, which is determined by the existence of corresponding MRD. In case that the MRD does not exist, the following procedure should be performed:

- (1) Remove the MPV asset file.
- (2) Remove the simple MPV asset description that points to the MPV asset file in the MPV manifest. (E.g., the asset manifest file in the AR directory)

If the corresponding MRD exist, the following procedure should be performed:

- (1) Extract the list of reference links in the MRD.
- (2) Remove the reference link of the simple asset from the list, which is just selected to remove.
- (3) For one of the reference links that is left in the list, determine whether it should be removed or not. It could be automatically determined by design policy or user scenario.
- (4) If a certain reference link of a composite asset description which refers the simple MPV asset description, is determined to be removed, perform the composite asset remove procedure below.
- (5) Iterate steps from (3) to (4) for each of the reference links in the MRD.
- (6) By the removing condition of MRD, remove the MRD file if necessary.

Removing a composite MPV asset description in an MPV manifest is done by a set of remove operations of simple MPV asset descriptions. The corresponding MRD of each of the simple asset descriptions which is referred by a composite asset description should always exist. Thus, the procedure is as follows:

- (1) Extract the list of reference links in the MRD.
- (2) Remove the reference link of the composite asset from the list, which is just selected to remove.
- (3) For one of the reference links which is left in the list, determine if it should be removed or not. It could be automatically determined by design policy or user scenario.
- (4) If a certain reference link of a simple asset is determined to remove, perform the simple asset remove procedure above.
- (5) Iterate steps from (3) to (4) for each reference link in the MRD.
- (6) By the removing condition of MRD, remove the MRD file if necessary.

### **I.3.4 Update**

While browsing a manifest in an MPV manifest file, if the metadata corresponding to the asset description is needed to update, the related metadata may be synchronized. For example, the initial metadata has poor information in general, which is generated by predefined data or design policy. When an update of metadata of an asset occurs by design policy, user input or scenario, the other metadata of the asset can be updated by MRD if it exists, as follows:

- (1) Update the metadata of the asset description in the MPV manifest.
- (2) Extract the list of reference links in the corresponding MRD.
- (3) For one of the reference links that is in the list, determine if the metadata of it should be updated or not. It could be automatically determined by some policy or depending on some user scenarios.
- (4) If a certain asset is determined to be updated, perform “update” procedure on the asset that is referred by the reference link.
- (5) Iterate from step (3) to (4) for each of the reference links in the MRD.

### **I.3.5 Copy**

Perform the create operation for the destination manifest and directory.

### **I.3.6 Move**

Perform the create and remove operations for the manifests and directories of original and destination, respectively.

## **I.4 Example**

The following is a simple example of PS system using MRD.

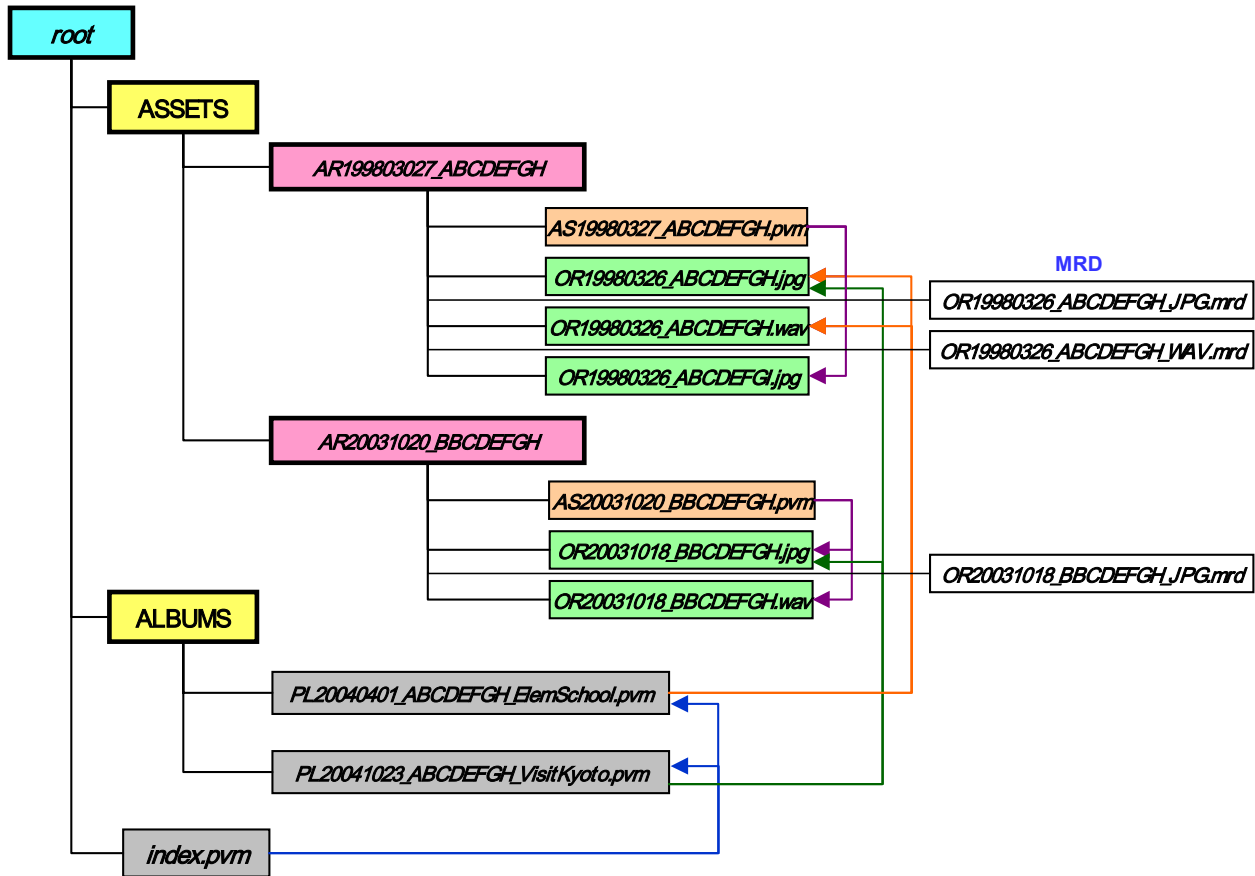


Figure I-1 An example of PS system using MRD

(1) A example of MPV manifest

```

<!--ALBUMS/PL20030401 ABCDEFGH ElemSchool.pvm-->
<file:Manifest>
  <mpv:AssetList>
    <mpv:Still mpv:id="MariahPhoto00">
      <LastURL>ASSETS/AR19980327_ABCDEFGH/OR20030326_ABCDEFGH.jpg</LastURL>
      <nmf:Metadata>
        <Properties xmlns="http://purl.org/dc/elements/1.1/">
          <title>Beautiful Mariah</title>
          <creator>Fan of Mariah</creator>
        </Properties>
        <Properties xmlns="http://purl.org/dc/terms/">
          <created>2004-04-20T05:00:03Z</created>
        </Properties>
      </nmf:Metadata>
    </mpv:Still>
    <mpv:Audio mpv:id="MariahSong00">
      <LastURL>ASSETS/AR19980327_ABCDEFGH/OR20030326_ABCDEFGH.wav</LastURL>
    </mpv:Audio>
    <mpv:StillWithAudio mpv:id="MariahAlbum00">
      <mpv:StillRef mpv:idRef=" MariahPhoto00"/>
      <mpv:AudioRef mpv:idRef=" MariahSong00"/>
    </mpv:StillWithAudio>
    ...
  </mpv:AssetList>
</file:Manifest>

```

## (2) Another example of MPV manifest

```
<!-- ALBUMS/PL20041013_ABCDEFGH_VisitKyoto.pvm -->
<file:Manifest>
  <mpv:AssetList>
    <mpv:Still mpv:id="MandyMoorePhoto00">
      <LastURL>ASSETS/AR20031020_ABCDEFGH/OR20031018_BBCDEFGH.jpg</LastURL>
      <nmf:Metadata>
        <Properties xmlns="http://purl.org/dc/elements/1.1/">
          <title>Cute Mandy</title>
          <creator>Fan of Mandy</creator>
        </Properties>
        <Properties xmlns="http://purl.org/dc/terms/">
          <created>2004-04-21T05:00:03Z</created>
        </Properties>
      </nmf:Metadata>
    </mpv:Still>
    <mpv:Still mpv:id="MariahSnapshot00">
      <LastURL>ASSETS/AR19980327_ABCDEFGH/OR20030326_ABCDEFGH.JPG</LastURL>
      <nmf:Metadata>
        <Properties xmlns="http://purl.org/dc/elements/1.1/">
          <title>Beautiful Mariah</title>
          <creator>Fan of Mariah</creator>
        </Properties>
        <Properties xmlns="http://purl.org/dc/terms/">
          <created>2004-04-20T05:00:03Z</created>
        </Properties>
      </nmf:Metadata>
    </mpv:Still>
    ...
  </mpv:AssetList>
</file:Manifest>
```

## (3) An example of MRD

```
<!-- asset/AR19980302_ABCDEFGH/OR20030326_ABCDEFGH.JPG.mrd -->
<?xml version="1.0"?>
<ps:ReferenceLink xmlns:ps="http://ns.osta.org/ps/1.0/">
  <ps:LastURL>AS19980327_ABCDEFGH.pvm?mpv_id=photo00000001</ps:LastURL>
  <ps:LastURL>../../album/PL20040401_ABCDEFGH_ElemSchool.pvm?mpv_id=MariahPhoto00</ps:LastURL>
  <ps:LastURL>../../album/PL20041023_ABCDEFGH_VisitKyoto.pvm?mpv_id=MariahSnapshot00</ps:LastURL>
</ps:ReferenceLink>
```

# Appendix II: 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).

## [EXIF-1998]

"EXIF", JEIDA standard, English Version 2.1, 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>

## [FreeDB]

"DISCID Howto", at <http://freedb.org/modules.php?name=Sections&sop=viewarticle&artid=6>



**[ID3]**

"The MPEG audio meta data format ID3", Martin Nielson, Dec. 1, 2000, Expired Internet Draft. Available at <http://www.watersprings.org/pub/id/draft-nilsson-id3-00.txt>

**[ID3-230]**

"ID3 Tag Version 2.3.0 ", Feb. 3, 1999, Martin Nielson, an informal standard.  
Available at <http://www.id3.org/id3v2.3.0.txt>

**[ID3-240]**

"ID3 Tag Version 2.4.0 - Native Frames", Nov. 1, 2000, Martin Nielson, an informal standard. Available at <http://www.id3.org/id3v2.4.0-frames.txt>

[ISO-639-2]

**[ISO639-1]**

'ISO/FDIS 639-1. Codes for the representation of names of languages, Part 1: Alpha-2 code.' Technical committee / subcommittee: TC 37 / SC 2

**[ISO3166-1]**

'ISO 3166-1 *Codes for the representation of names of countries and their subdivisions - Part 1: Country codes.*' Available at <http://www.iso.org/>

**[ISO8601]**

"Data elements and interchange formats - Information interchange - Representation of dates and times", International Organization for Standardization, 1998.  
Available at <http://www.iso.org/>

**[ISO10646]**

"Information Technology -- Universal Multiple-Octet Coded Character Set (UCS) -- Part 1: Architecture and Basic Multilingual Plane", ISO/IEC 10646-1:1993.  
Available at <http://www.iso.org/>

**[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>

#### **[MIMEMPA]**

"RTP Profile for Audio and Video Conferences with Minimal Control", H. Schulzrinne and others, anuary 1996, RFC 1890.

Available at <http://www.ietf.org/rfc/rfc3003.txt>. Relevant discussion in [MIMEMPEG].

#### **[MIMEMPEG]**

"The audio/mpeg Media Type", Martin Nilsson, November 2000, RFC 3003.

Available at <http://www.ietf.org/rfc/rfc3003.txt>

#### **[MIMEMPEG4]**

"RTP payload format for MPEG-4 Audio/Visual streams", Y. Kikuchi, T. Nomura, S. Fukunaga, Y. Matsui, H. Kimata, Nov. 2000, RFC 3016, Internet Engineering Task Force.

Available at <ftp://ftp.isi.edu/in-notes/rfc3016.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.

**[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/>