

---

---

**Information technology — Multimedia  
application format (MPEG-A) —**

Part 9:  
**Digital Multimedia Broadcasting  
application format**

AMENDMENT 1: Conformance and  
reference software

*Technologies de l'information — Format pour application multimédia  
(MPEG-A) —*

*Partie 9: Format pour application de diffusion générale multimédia  
numérique*

*AMENDEMENT 1: Logiciel de référence et conformité*

**PDF disclaimer**

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.



**COPYRIGHT PROTECTED DOCUMENT**

© ISO/IEC 2010

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office  
Case postale 56 • CH-1211 Geneva 20  
Tel. + 41 22 749 01 11  
Fax + 41 22 749 09 47  
E-mail [copyright@iso.org](mailto:copyright@iso.org)  
Web [www.iso.org](http://www.iso.org)

Published in Switzerland

## Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Amendment 1 to ISO/IEC 23000-9:2008 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23000-9:2008/Amd 1:2010

# Information technology — Multimedia application format (MPEG-A) —

## Part 9: Digital Multimedia Broadcasting application format

### AMENDMENT 1: Conformance and reference software

Page 33, after Table 7

Add the following two new clauses.

## 8 Conformance

### 8.1 Introduction

This Clause describes the conformance suite for the DMB-AF standard. The purpose of the conformance suite is to cover the valid features that may be exercised in terms of storage and playback of various DMB contents.

The conformance suite has several aspects: file conformance, player conformance, and creator conformance. Each conformance is defined in the subsequent Subclauses.

Several conformance files are also provided as associated files with this standard. By using the conformance files together with the reference software, most major features of DMB-AF can be exercised and checked.

### 8.2 File conformance

#### 8.2.1 File conformance in general

A file conforms to a brand of DMB-AF when the following conditions are all satisfied;

- Brand consistency: the major\_brand field of 'ftyp' box matches the declared brand name.
- File format conformance: the structure and syntax of the boxes included in the file conform to the box types and structure defined in Table 3 of 6.1.
- Component consistency: all the components contained in the file can be found in the brand entry of Table 5, Table 6, or Table 7 in Clause 7.
- Component conformance: each component in the file such as a media stream or an xml document conforms to the corresponding specification and restrictions listed in Table 1. In addition, the conditions defined in 8.2.2 to 8.2.9 are also satisfied.

When a file conforms to a brand of DMB-AF, the file can be called as DMB-AF file.

## 8.2.2 DAB and DAB+ stream storage

### 8.2.2.1 Introduction

DAB (ETSI EN 300 401) audio service is provided by using MPEG Audio Layer II compression and the fixed-length encapsulation of the compressed audio into "DAB audio frames". DAB audio frames may include associated auxiliary data called PAD (Program-Associated Data). The PAD area is composed of a fixed-length PAD (F-PAD) and a variable-length PAD (X-PAD). Through the X-PAD area, auxiliary data associated with the audio can be delivered in a synchronous or in an asynchronous fashion. Representative of such associated services are DLS (Dynamic Label Service) and SLS (Slide Show Service). A dynamic label in DLS delivers a short text message informing the station name, the title and/or singer's name of the on-air music, any message related or not related to the main audio service, etc. Some still images or a sequence of still images can be delivered through the slide show service. Those images can be related to the main audio service such as album cover of the on-air music. Similarly the images can be the ones that are not related to the main audio service. Synchronization between the main audio and the dynamic label and/or the slides is naturally achieved since these auxiliary data is placed at the end of each audio frames that include the compressed audio to which the data is to be synchronized, i.e., to be played together.

DAB+ (ETSI TS 102 563) is an enhanced audio service more efficient than DAB. By using MPEG-4 Audio HE-AAC v2 instead of MPEG Audio Layer II, DAB+ can achieve more bitrate-efficient audio compression. In DAB+, five consecutive DAB audio frames are grouped to hold one DAB+ audio super frame which includes multiple (2 to 6) HE-AAC v2 audio access units (AUs). The particular number of AUs depends on the sampling frequency and the use of SBR (Spectral Band Replication). The auxiliary data encoded as the same way as the PAD of DAB may be included in the first part of each AU. The DLS and SLS can be provided in DAB+ through the auxiliary data area. For DAB+ streams, it is assumed that the decoding of parity bytes due to the Reed-Solomon code is performed with possible corrections before the storage and only the resulting sequence of DAB+ audio super frame data stream is stored within a DMB-AF file.

### 8.2.2.2 Sample format and random-access points

Sample definitions are determined according to the existence of `stss` (sync sample table) box. In case `stss` is absent, which means every samples are random-accessible, a sample is defined as an integral multiple of consecutive DAB audio frames or DAB+ audio super frames. In case `stss` box exists, then the whole DAB audio frame stream or the whole DAB+ audio super frame stream shall be treated as a single sample. In this case, the `entry_count` in the `stss` box shall take the value of 0, which means that the sync sample positions are unknown.

### 8.2.2.3 Sample description format

#### 8.2.2.3.1 Syntax

```
class DABSampleEntry() extends HintSampleEntry(name) {
    uint(16) hinttrackversion = 1;
    uint(16) highestcompatibleversion = 1;
    uint(8) pad_type;
    if (name == rdap) {
        uint(16) audio_super_frame_size;
    }
}
```

#### 8.2.2.3.2 Semantics

`name` is either `'rdab'` or `'rdap'`. In case of `'rdab'`, the stored samples are DAB audio frames. In case of `'rdap'`, the stored samples are DAB+ audio super frames.

`hinttrackversion` is currently 1; the `highestcompatibleversion` field specifies the oldest version with which this track is backward-compatible.

`pad_type` indicates the existence of PAD in the DAB audio frame stream or the DAB+ audio super-frame stream. The following values are defined for `pad_type`:

'0x00' : PAD type unknown. PAD may or may not exist in the stream.

- '0x01' : Only MOT-SLIDE data exists in the stream.
- '0x02' : Only DLS data exists in the samples.
- '0x03' : Both MOT-SLIDE and DLS data exists in the samples.
- '0x04'~'0xFF' : reserved for future use.

audio\_super\_frame\_size represents the DAB+ audio super frame size in bytes. For details, refer to ETSI TS 102 563.

### 8.2.3 T-DMB TS storage

The MPEG-4 SL (Synchronization Layer) over MPEG-2 TS (Transport Stream) structure is used in T-DMB TS. The T-DMB TS is directly stored within a DMB-AF file. It is assumed that the decoding of 16 parity bytes due to the Reed-Solomon code RS(204, 188) is performed with possible corrections before the storage and only the resulting sequence of 188-byte MPEG-2 TS packets are stored within a DMB-AF file. The details of the storage format are specified in 6.5 of this Part of ISO/IEC 23000.

### 8.2.4 S-DMB TS storage

The MPEG-4 SL (Synchronization Layer) over MPEG-2 TS (Transport Stream) structure may or may not be used in S-DMB TS. The S-DMB TS is directly stored within a DMB-AF file similarly to the T-DMB TS case. It is assumed that the decoding of 16 parity bytes in the Reed-Solomon code RS(204, 188) is performed with possible corrections before the storage and only the resulting sequence of 188-byte MPEG-2 TS packets are stored within a DMB-AF file. The details of the storage format are specified in 6.5 of this Part of ISO/IEC 23000.

### 8.2.5 MPEG-4 media storage

#### 8.2.5.1 Introduction

Among the MPEG-4 media that the DMB-AF supports are ER-BSAC, HE-AAC2, AVC, AAC+SBR, and BIFS. When BIFS is used for interactive data contents, JPEG or PNG may also be used. ER-BSAC compression specified in ISO/IEC 14496-3 is used without the error-resilience functionality. Also some additional restrictions shall apply according to ETSI TS 102 428. HE-AAC2 compression is used in T-DMB Profile 2 instead of ER-BSAC. It is also used in DAB+ instead of MPEG Audio Layer II. The further restrictions are specified in ETSI TS 102 428 for T-DMB Profile 2 and ETSI TS 102 563 for DAB+, respectively. AAC+SBR audio for S-DMB consists of MPEG-2 AAC LC Profile and MPEG-4 Audio SBR. Further restrictions on AAC+SBR are specified in TTAS.KO-07.0027.

The particular AVC profile and level used in this Part of ISO/IEC 23000 are Baseline Profile and Level 1.3. In addition, several restrictions shall apply according to ETSI TS 102 428. The particular BIFS scene description profile and level used in this specification are Core2D and Level 1, respectively. The particular BIFS graphics profile and level used in this specification are Core2D and Level 1, respectively. In addition, several restrictions shall apply according to ETSI TS 102 428.

Three different storage types are allowed in this specification:

- MPEG-2 TS encapsulated case
- Separate ES case without MPEG-4 Systems context
- Separate ES case with MPEG-4 Systems context

#### 8.2.5.2 MPEG-2 TS encapsulated case

In this case, the MPEG-4 media are encapsulated by MPEG-2 TS. The same rule specified in 8.2.3 is applied.

#### 8.2.5.3 Separate ES case without MPEG-4 Systems context

When a DMB-AF file only includes audio contents encoded by ER-BSAC, HE-AAC2, or AAC+SBR, a single track for each audio content is allocated. Each of the audio ESs is divided into audio samples. A single audio AU or a consecutive multiple audio AUs form a sample. The MP4AudioSampleEntry('mp4a') as specified in ISO/IEC 14496-14 is used for the description of the audio samples.

When a DMB-AF file includes an AVC video with an associated audio encoded with ER-BSAC, HE-AAC2, or AAC+SBR, two tracks are allocated: one for AVC video data and the other for the audio data. The AVC video ES is divided into video samples. The AVCSampleEntry('avc1') as specified in ISO/IEC 14496-15 is used for the description of the video samples. For the associated audio, the same rule as the previous paragraph applies.

#### 8.2.5.4 Separate ES case with MPEG-4 Systems context

A video content can be accompanied by an associated interactive data content based on BIFS. Also an audio-centric content can be accompanied by an associated video and/or interactive data content based on BIFS. In these cases, the storage of the contents into a DMB-AF file requires MPEG-4 Systems context. The required data in the context of MPEG-4 Systems such as IOD, OD ES, and BIFS ES shall be stored together with the audio and video content in a DMB-AF file. According to ISO/IEC 14496-14, IOD is stored in 'iods' box, OD ES and BIFS ES are stored as separate tracks with MpegSampleEntry('mp4s'), AVC video ES and audio ES are stored as separate tracks with MP4VisualSampleEntry('mp4v') and MP4AudioSampleEntry('mp4a'), respectively.

#### 8.2.6 Image storage (JPEG, PNG, MNG)

##### 8.2.6.1 Introduction

Images can be stored as separate items as specified in ISO/IEC 14496-12.

##### 8.2.6.2 Image item storage

Images can be stored as item if no time-synchronization is required. In this case, image items shall be registered by their own CRIDs in the TVA metadata in order that they can be browsed from a TVA browser. The CRIDs shall also be registered into 'crid' box in order for the DMB-AF player to correctly relate each CRID with the corresponding item\_id and hence with the physical location in 'mdat' box.

##### 8.2.6.3 Image track storage

Images can also be stored as track if a timely playback is required (e.g., slideshow). In this case, images are stored as media tracks.

#### 8.2.7 3GPP Timed Text storage

3GPP timed text shall be stored as a text track with text sample entry 'tx3g'. The text samples stored in the 'mdat' box and the boxes describing the synchronization information of the text samples shall conform to 3GPP TS 26.245.

#### 8.2.8 NPAD (Non-Program Associated Data) storage

##### 8.2.8.1 Introduction

The DMB-AF specification enables storing NPAD contents as non-movie items. The information about the data is stored in the file format by its location and size inside the media data box. The NPAD contents should be extracted from the DMB-AF file and then executed by an external application or an application embedded in the DMB-AF player according to the NPAD content type.

##### 8.2.8.2 BWS storage

For a BWS package, the information about the relative path of each resource of BWS contents is stored in the item information ('inf') box of the DMB-AF file. Furthermore, the crid and item location information is also stored. The BWS contents to be stored in to DMB-AF file shall already be contained in a single directory that links to each other in the main web page file. When the DMB-AF player need to extract the BWS contents, it reads the registered list of BWS contents in the crid box, locates the physical file, and calls BWS browser to open the main web page file.

### 8.2.8.3 Java MIDlet storage

For a Java MIDlet package, the information about the relative path of the compressed package (in 'zip' format) of Java MIDlet contents is stored in the item information ('iinf') box. Furthermore, the crid and item location information. The Java MIDlet contents to be packaged into DMB-AF file shall be already compressed into a zip file. When the DMB-AF player needs to extract the Java MIDlet content, it reads the midlet information in crid box, locates the physical midlet data, unzips the midlet data and finally calls a midlet player to execute the midlet.

### 8.2.9 Metadata storage (DID, IPMP, REL, TVA)

Metadata stored inside a DMB-AF file shall be annotated according to the structure specified in this Subclause.

The metadata shall be encapsulated in form of MPEG-21 DID as the root container for other types of metadata allowed in this specification. The components of DMB-AF are annotated in the TV-Anytime metadata, therefore Item elements in the DID is not used, unless the components are protected and the protection is signaled using MPEG-21 IPMP as described below.

In the necessity to use the TV-Anytime metadata to annotate the content description, the metadata shall be annotated as Statement in a Descriptor element of MPEG-21 DID.

In the necessity to use the MPEG-21 IPMP to annotate the protection and governance of the components of DMB-AF, the IPMP General Info metadata shall be annotated as Statement in a Descriptor element of DID. The Descriptor/Statement for IPMP shall be different one that used for TV-Anytime metadata. If used, the IPMP DIDL metadata shall be annotated for all the components for which protection and governance are signaled by the IPMP. The protection tool used to protect the protected component is referred to IPMP General Info using IPMP DIDL Identifier. The protected component is referred to the physical location in file format via crid functionality, thus the reference annotated in the IPMP DIDL Contents shall be annotated as the component's crid identifier.

In the necessity to use the MPEG-21 REL to annotate the license information of the components of DMB-AF, the REL metadata shall be annotated according to the existence of MPEG-21 IPMP within the MPEG-21 DID structure. If MPEG-21 IPMP is exist, the REL metadata shall be annotated as License in IPMPInfo License Collection. Otherwise, the REL metadata shall be annotated as Statement in a Descriptor element different to that of TV-Anytime metadata.

Definition of namespaces for different types of metadata can be defined in the DIDL root element of MPEG-21 DID or can be defined separately in each of metadata type root element. It is encouraged to validate the metadata to be stored in the DMB-AF against the schema prior to be added to the DMB-AF.

Figure 16 illustrates the structure of metadata.



Figure 16 — (a) Structure of DID metadata in the presence of IPMP, REL and TVA; (b) without the presence of IPMP; (c) only TVA

### 8.3 Player conformance

A file player conforms to a brand of DMB-AF when the following condition is satisfied;

- The player can correctly parse any file that conforms to the brand and can appropriately play all the components included in the file.

When a file player conforms to a brand of DMB-AF, the player can be called as a DMB-AF player.

#### 8.4 Creator conformance

A file creator (or authoring tool) conforms to a brand of DMB-AF when the following condition is satisfied:

- The creator can produce files that conform to the brand.

When a file creator (or authoring tool) conforms to a brand of DMB-AF, the creator (or authoring tool) can be called as a DMB-AF creator (or authoring tool).

#### 8.5 Conformance files

The following table describes about the conformance files associated to this standard:

File name		01_T-DMB-1_TS.dmb
Description		a music show program (T-DMB Profile 1) with descriptive metadata and caption text
Major brand		dv1b
Compatible brands		dmb1, dv1a
Components Inside the file	An MPEG-2 TS (containing ER-BSAC and AVC)	- stored as an 'm2ts' hint track
	A DID XML (containing TVA and REL)	- each image is stored as an item - a TVA for content description and segment information - a REL for defining a simple license
	14 JPEG images related to TVA	- a program title image - a segment group title image - 12 segment title images
	A 3GPP-TT stream	- caption text for the program

File name		02_T-DMB-1_AV.dmb
Description		the same as file '01'
Major brand		dv1b
Compatible brands		dmb1, dv1a, avc1, mp42
Components Inside the file	AVC, ER-BSAC	- each stream is stored as separate track - AVC as AVCSampleEntry('avc1') - ER-BSAC as MP4AudioSampleEntry('mp4a')
	A DID XML (containing TVA and REL)	- a TVA for content description and segment information - a REL for defining a simple license
	14 JPEG images related to TVA	- each image is stored as an item - a program title image - a segment group title image - 12 segment title images

File name		03_T-DMB-1_BIFS(TS).dmb
Description		a golf program with BIFS multiplexed in TS (T-DMB Profile 1)
Major brand		dv1a
Compatible brands		dmb1, dv1b
Components Inside the file	An MPEG-2 TS (containing ER-BSAC, AVC, OD, BIFS and JPEG images)	- stored as an 'm2ts' hint track

File name		04_T-DMB-1_BIFS(ES).dmb
Description		a golf program with BIFS in a separate track
Major brand		dv1a
Compatible brands		dmb1, dv1b, avc1, mp42

Components Inside the file	AVC, ER-BSAC, OD, BIFS and JPEG images	<ul style="list-style-type: none"> <li>- each stream is stored as separate a track</li> <li>- AVC as MP4VisualSampleEntry('mp4v')</li> <li>- ER-BSAC as MP4AudioSampleEntry('mp4a')</li> <li>- IOD as ObjectDescriptorBox('iods')</li> <li>- OD as MpegSampleEntry('mp4s')</li> <li>- BIFS as MpegSampleEntry('mp4s')</li> <li>- JPEG as MP4VisualSampleEntry('mp4v')</li> </ul>
----------------------------	--	--

<b>File name</b>		<b>05_T-DMB-1_IPMP(TS).dmb</b>
Description		the same as file '01'
Major brand		dv1b
Compatible brands		dmb1
Components Inside the file	An MPEG-2 TS (containing ER-BSAC and AVC)	<ul style="list-style-type: none"> <li>- stored as an 'm2ts' hint track (the same TS as is in the conformance file 01).</li> <li>- the TS is encrypted with AES-128 CTR (key: "key1")</li> </ul>
	A DID XML (containing TVA, REL and IPMP)	<ul style="list-style-type: none"> <li>- a TVA for content description and segment information (the same TVA as is in the conformance file 01)</li> <li>- an REL for defining a simple license (the same TVA as in the conformance file 01)</li> <li>- a IPMP for describing the protection tool</li> </ul>
	14 JPEG images related to TVA	<ul style="list-style-type: none"> <li>- each image is stored as an item</li> <li>- a program title image</li> <li>- a segment group title image</li> <li>- 12 segment title images (one of the images is protected with the same tool and key as for the TS)</li> </ul>

<b>File name</b>		<b>06_T-DMB-1_IPMP(ES).dmb</b>
Description		the same as file '02'
Major brand		dv1b
Compatible brands		dmb1
Components Inside the file	AVC, ER-BSAC	<ul style="list-style-type: none"> <li>- each streams is stored as separate track</li> <li>- Each ES is encrypted with AES-128 CTR (key: "key1")</li> </ul>
	A DID XML (containing TVA, REL and IPMP)	<ul style="list-style-type: none"> <li>- a TVA for content description and segment information (the same TVA as is in the conformance file 02)</li> <li>- an REL for defining a simple license (the same TVA as in the conformance file 02)</li> <li>- an IPMP for describing the protection tool</li> </ul>
	14 JPEG images related to TVA	<ul style="list-style-type: none"> <li>- each image is stored as an item</li> <li>- a program title image</li> <li>- a segment group title image</li> <li>- 12 segment title images (one of the images is protected with the same tool and key as for the ESS)</li> </ul>

<b>File name</b>		<b>07_DAB_PAD.dmb</b>
Description		a DAB music program with lyric and slideshow
Major brand		da0b
Compatible brands		dmb1, da0a
Components Inside the file	A DAB Audio Frames Stream (containing MPEG-1 Audio Layer II, DL, and MOT Slides)	<ul style="list-style-type: none"> <li>- stored as an 'rdab' hint track</li> <li>- DL text and MOT slides reside in the PAD area of the DAB audio frames</li> </ul>
	A DID XML (containing TVA)	<ul style="list-style-type: none"> <li>- a TVA for content description and segment information</li> </ul>
	3 JPEG images related to TVA	<ul style="list-style-type: none"> <li>- each image is stored as an item</li> <li>- a program title image</li> <li>- 2 segment title images</li> </ul>

<b>File name</b>		<b>08_DAB_Plus.dmb</b>
Description		a DAB+ music program
Major brand		da2a
Compatible brands		dmb1, da2b, mp42
Components Inside the file	A DAB+ Audio Frames Stream (containing HE-AAC2)	<ul style="list-style-type: none"> <li>- stored as an 'rdap' hint track</li> </ul>

<b>File name</b>		<b>09_T-DMB-2_TS.dmb</b>
Description		a T-DMB Profile 2 program (Visual Radio)
Major brand		dv2a
Compatible brands		dmb1, dv2b
Components Inside the file	AVC, HE-AAC2	- stored as an 'm2ts' hint track

<b>File name</b>		<b>10_T-DMB-2_AV.dmb</b>
Description		a T-DMB Profile 2 program (Visual Radio)
Major brand		dv2a
Compatible brands		dmb1, dv2b, avc1, mp42
Components Inside the file	AVC, HE-AAC2	- each stream is stored as separate track

<b>File name</b>		<b>11_MIDLET.dmb</b>
Description		a sokoban (warehouse keeper) game program
Major brand		dmb1
Compatible brands		none
Components Inside the file	Java MIDlet package	- the package is compressed with Zip and stored as a single item
	A DID XML (containing TVA)	- a TVA for content description for the game program
	1 JPEG images related to TVA	- a title image stored as an item

<b>File name</b>		<b>12_T-DMB-1_BWS.dmb</b>
Description		a music show program (T-DMB Profile 1) with descriptive metadata and BWS web page
Major brand		dmb1
Compatible brands		dv1b, dv1a
Components Inside the file	An MPEG-2 TS (containing ER-BSAC and AVC)	- stored as an 'm2ts' hint track (the same TS as is in the conformance file 01)
	A DID XML (containing TVA)	- a TVA for content description and segment information
	15 JPEG images related to TVA	- each image is stored as an item - a program title image - a BWS title image - a segment group title image - 12 segment title images
	BWS resources	- html, css, and PNG images for BWS - stored as separate items

<b>File name</b>		<b>13_S-DMB_AV.dmb</b>
Description		an entertainment program preview
Major brand		dv3a
Compatible brands		dv3b, dmb1
Components Inside the file	AVC, AAC+SBR	- each stream is stored as separate track - AVC as AVCSampleEntry('avc1') - AAC+SBR as MP4AudioSampleEntry('mp4a')

The following table shows a summary of conformance files in terms of contained media and metadata components.

Table 8 — Summary of conformance files

File name	Description	Brands		Audio				PAD	Video	TS	BIFS	Image	Text	Metadata				NPAD		
		major	compatible	MPEG L2	ER-BSAC	HE-AAC2	AAC+SBR	DLS	MOT Slide	AVC	M2TS	BIFS	JPG/PNG/MNG	3GPP-TT	DID	TVA	REL	IPMP	BWS	MIDLET
01_T-DMB-1_TS.dmb	a music show program (T-DMB Profile 1) with TVA/REL and caption text	dv1b	dmb1, dv1a		*															
02_T-DMB-1_AV.dmb	same as 01 but ESs are stored as separate tracks	dv1b	dmb1, dv1a, avc1, mp42																	
03_T-DMB-1_BIFS(TS).dmb	a golf program with BIFS multiplexed in TS (T-DMB Profile 1)	dv1a	dmb1, dv1b		*				*		*									
04_T-DMB-1_BIFS(ES).dmb	same as 03 but ESs are stored as separate tracks	dv1a	dmb1, dv1b, avc1, mp42																	
05_T-DMB-1_IPMP(TS).dmb	same as 01 but encrypted with "key1"	dv1b	dmb1		*				*											
06_T-DMB-1_IPMP(ES).dmb	same as 02 but encrypted with "key1"	dv1b	dmb1																	
07_DAB_PAD.dmb	a DAB music program with lyric and slideshow	da0b	dmb1, da0a, mp42																	
08_DAB_Plus.dmb	a DAB+ music program	da2a	dmb1, da2b, mp42																	
09_T-DMB-2_TS.dmb	a T-DMB Profile 2 program (Visual Radio)	dv2a	dmb1, dv2b		*				*											
10_T-DMB-2_AV.dmb	a T-DMB Profile 2 program (Visual Radio)	dv2a	dmb1, dv2b, avc1, mp42																	
11_MIDLET.dmb	a sokoban game (java midlet) with TVA	dmb1	none																	
12_T-DMB-1_BWS.dmb	a music show program with TVA and BWS data	dmb1	dv1b, dv1a		*				*											
13_S-DMB_AV.dmb	a music show program with TVA and BWS data	dv3a	dv3b, dmb1				*		*											

Note 1	O	The corresponding component exists in the file
	*	The corresponding component is multiplexed in the TS
Note 2	T-DMB-1	T-DMB Profile 1, i.e., with ER-BSAC audio
	T-DMB-2	T-DMB Profile 2, i.e., with HE-AAC v2 audio
Note 3	BIFS(TS)	BIFS ES is multiplexed in TS
	BIFS(ES)	BIFS ES is separately stored in track
Note 4	IPMP(TS)	TS is protected
	IPMP(ES)	Each ES's are separately stored and protected

## 9 Reference Software (RSW)

### 9.1 Introduction

The DMB-AF RSW consists of three application programs: the DMB-AF Packager, Media Player, and Metadata Browser. The Packager provides one possible implementation for packaging the DMB contents and metadata into a single DMB-AF file. The Media Player provides one possible implementation for parsing DMB-AF files and playing the included timed media components. The Metadata Browser provides browsing of untimed metadata and data components included in DMB-AF files, for example, TVA, REL, and JPMP metadata as well as images, BWS, and Java MIDlet.

The following Subclauses describe the functional model and architecture of the reference software for DMB-AF, where some design concept and general architecture of the software are provided. For the exact implementation details, the software itself should be consulted.

### 9.2 Composition of the software

The directory structure of the RSW is as follows:

Folder name	Description
bin	Contains the executables and DLLs.
inc	Contains the header files for the libraries.
lib	Contains the library files.
proj	Contains the three applications and library source codes.
tools	Contains auxiliary tools such as file parser and converters.

Within the 'proj' folder, there are three application subfolders and two library subfolders, each of which is summarized in the following table:

Folder name	Sub-folder name and/or Description	Applications using the library (O: used, X: not used)		
1_Packager	<i>Packager project folder. (Application)</i>			
2_MediaPlayer	<i>Media Player project folder. (Application)</i>			
3_MetadataBrowser	<i>Metadata Browser project folder. (Application)</i>			
4_Libraries	<i>Libraries for the above three application projects.</i>	Packager	Media Player	Metadata Browser
	AAC_Dec	X	O	X
	AVC_Dec	X	O	X
	BSAC_Dec	X	O	X
	Crypt_Lib	O	X	O
	CryptoPP_Lib	O	X	O
	CxImage_Lib	O	X	X
	DMB_ISO_Lib	O	X	X
	GDIplus	X	O	X
	ISO_Lib	O	O	X
	JPEG_Dec	O	O	O
	MATEMon_Lib	X	X	O
	ZipArchive_Lib	X	X	O
	_msxml	O	X	O
5_PrivateLibraries	<i>Libraries for the above three application projects.</i>			
	ColorSpaceConverter_Lib	X	O	X
	DABAudioDecoder_Lib	X	O	X

DID_Lib	<i>DID generator library and header. In the '_doc' folder of this library, API guide is included.</i>	O	X	O
DMB_MATE_RI2	<i>DMB MATE emulator</i>	X	X	O
REL_Lib	<i>REL parser library and header.) In the '_doc' folder of this library, API guide is included.</i>	O	X	O
TS_Demux	<i>MPEG- TS Demux library and header. In the '_doc' folder of this library, API guide is included.</i>	X	O	X
TT_Lib	<i>3GPP Timed-text generator library and header. In the '_doc' folder of this library, API guide is included.</i>	O	O	X
TVA_Lib	<i>TVA parser library and header. In the '_doc' folder of this library, API guide is included.</i>	O	X	O

The Packager, Media Player, and Metadata Browser folders contains corresponding Microsoft VC++ 6.0 project workspace and source codes, while the subfolders of the Libraries folder may contain only the libraries and header files with or without source codes.

### 9.3 Packager

#### 9.3.1 Introduction

DMB-AF Packagers, in general, create DMB-AF files conforming to the brands defined in Clause 7. The current version of Packager can only handle the components package MPEG-2 TS file with TVA and REL metadata. It can also accept encapsulated inside MPEG-21 DID and 3GPP timed text. It creates the DMB-AF file by packaging well-prepared bitstreams into the ISO-base file format. This means the packager cannot be used to create the content such as creating AVC bitstreams or creating the TVA/DID metadata.

#### 9.3.2 Software architecture

The DMB-AF packager has the functional modules as shown in Figure 17. The description of each module is described in Table 9.

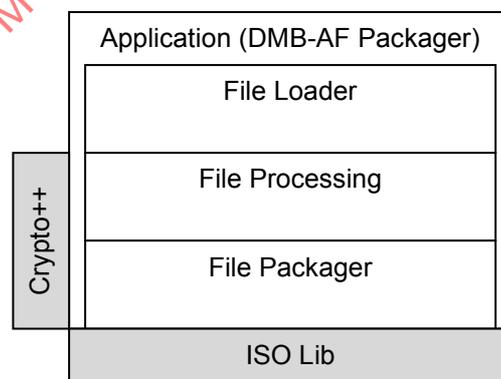


Figure 17 — Logical architecture of functional module of DMB-AF packager

Table 9 — List of functional modules in the DMB-AF packager

No.	Function	Type	Description
1	File loader	Internal	Load the files to be packaged into DMB-AF file
2	File processing	Internal	Encrypt protected content; Create 3GPP Timed Text structure. It consists of several modules as described in 9.3.3
3	File packager	Internal	Collect the list of files that already loaded using the file loader and package the file into file format. It consists of several modules as described in 9.3.3
4	ISOLib	External	Package the TS file into MP4 file format.
5	Crypto++	External	Provides algorithm for encryption for protected contents

9.3.3 Functional description

The DMB-AF packager has the software architecture as shown in Figure 18. The description of each module is described in Table 10.

Note that there are two ways of creating a DMB-AF file using the Packager;

- two step method: create DMB-AF file with media resources first, then open the DMB-AF file and add metadata to the file.
- one step method: create DMB-AF file with media resources and metadata in turn, at a time.

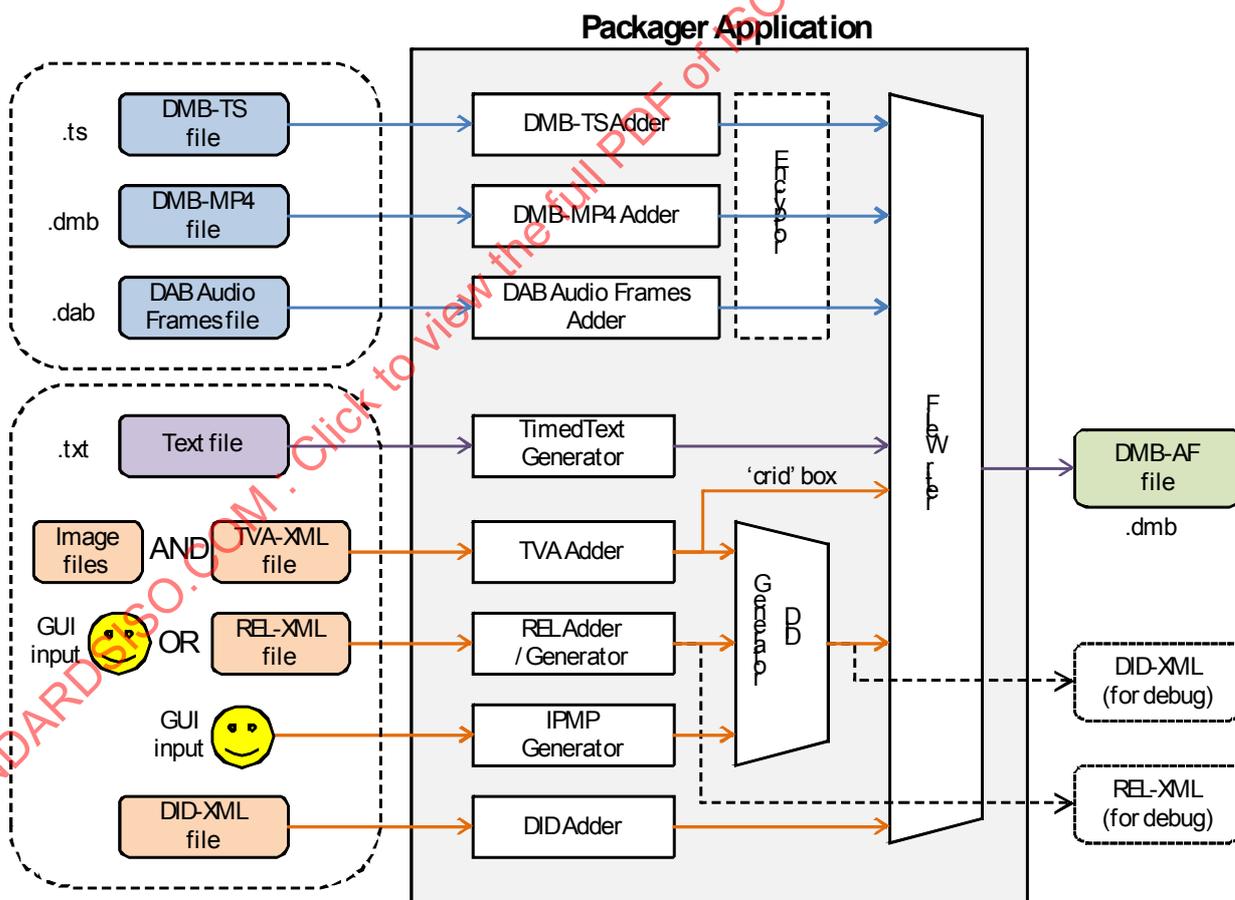


Figure 18 — Software architecture and I/O interface of DMB-AF packager

**Table 10 — List of implementation modules in the DMB-AF packager**

Module	Functionality	Description	Implementation
File loader	Add TS	Add TS file into the packager	Addts()
	Add TVA XML	Add TVA XML file into the packager	Addtva()
	Define IPMP	Define IPMP General Information	AddIPMP()
	Add/Define REL XML	Add REL XML file into the IPMP definition, or manually annotated	AddREL()
	Add Text	Add plain text file with simple synchronization information into the packager and convert into timed text packages	Addtext()
	Add BWS	Add BWS contents into the packager	AddBWS()
	Add Java MIDlet	Add zipped Java MIDlet into the packager	AddJavaMIDlet()
File processing	Encryptor	Encrypt protected resources	CTR_Crypt()
	3GPP Creator	Generate timed text stream by creating sample table box and sample text for timed text	initTTGen() createTT()
File packager	DID Creator	Generate DID XML data by inserting TVA and REL XML. It also creates the META box for the DMB file format	createDIDFile() createMetaFile()
	M2TS Handler	Modifies MP4 file created by ISOLib by adding the required M2TS handler box	updatehandler()
	MP4 Extractor/ Repackager	Extract MP4 boxes from MP4 file and 3GPP file, and repackage the boxes with respective updates into a DMB file format	MP4ExtractToFile() MP4ExtractTrakfromMOOV() MP4ReadFile() MP4RepackageFile() MP4UpdateMDAT() MP4UpdateMETA() MP4UpdateMOOV() MP4UpdateNewTRAK() MP4UpdateOffset()
ISOLib	ISOLib	Package TS file into MP4 file format	As described in reference
Crypto++	Crypto++	Provides encryption algorithm	As described in reference

**9.4 Media Player**

**9.4.1 Introduction**

The DMB-AF player plays the DMB-AF file conforming to the 'dv2a' brand and 'dv3a' brand with 3GPP Timed Text and TV Anytime (TVA) metadata. It renders the audio and video as a movie (the video is synchronized with the audio), with timed text embedded within the screen or in the text box, and the corresponding TVA metadata that can be used to browse the DMB-AF content, jump to segment of the video, and show the usage history (as for the current version of the RSW, only already existing usage history can be shown, which means the player is not recording the user action when users perform any action to the content in the player).

**9.4.2 Software architecture**

The DMB-AF media player has the functional modules as shown in Figure 19. The description for each module is described in Table 11.

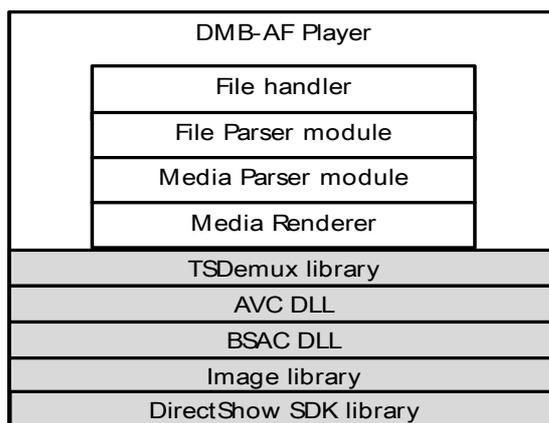


Figure 19 — Functional modules of DMB-AF media player

Table 11 — List of functional modules in the DMB-AF media player

No.	Function	Type	Description
1	File handler	Internal	Read data from DMB-AF files
2	File Parser module	Internal	Parse DMB-AF files and the boxes in the files
3	Media Parser module	Internal	Parse media data and decode the media data
4	Media Renderer	Internal	Initialize DirectShow and render media
5	TSDemux library	External	Perform demultiplexing of MPEG-2 TS encapsulating MPEG-4 systems data that conforms to T-DMB restrictions
6	AVC DLL	External	Decode MPEG-4 AVC data that conforms to T-DMB restrictions
7	BSAC DLL	External	Decode MPEG4 BSAC data that conforms to T-DMB restrictions
8	Image library	External	Decode image
9	DirectShow SDK library	External	Render audio-visual data

#### 9.4.3 Functional description

The DMB-AF media player has the software architecture as shown in Figure 20. The description of each module is described in Table 12.

**Error! Objects cannot be created from editing field codes.**

Figure 20 — Software architecture of DMB-AF media player

**Table 12 — List of implementation modules in the DMB-AF player**

Module	Functionality	Type	Description	Function
File Parser module	DMB-AF Parser	Internal	Parse boxes in a DMB-AF file	Run() parseAFFile()
Media Parser module	TSDemux	Internal	Extract BIFS, OD, Video, and Audio PESes from MPEG-2 TS data	Init() Start() Stop() MainThread() Run()
	PESParser	Internal	Extract AUs from PES data	Init() Run()
	BIFS Parser	Internal	Parse BIFS data	Init() Run()
	ODParser	Internal	Parse OD data	Run()
	MPEG-4 AVC Decoder	Internal	Decode MPEG-4 AVC Video stream	Init() Run()
	MPEG-4 BSAC Decoder	Internal	Decode MPEG-4 BSAC Audio stream	Init() Run()
	Image Decoder	Internal	Decode JPEG image stream	Run()
	DAB Decoder	Internal	Decode MPEG Layer2 Audio and PAD services	Run() DecodeFrame()
	Video Buffer	Internal	Hold video PUs	Init() Run()
	Audio Buffer	Internal	Hold audio PUs	Init() Run()
	Image Buffer	Internal	Hold image data	Start() Stop() Run()
Media Renderer	Presenter	Internal	Present AV scenes	Init() Start() Terminate() Render()
	Video Renderer	Internal	Render video using DirectShow SDK library	Create() Init() Clear() Paint() Terminate()
	Audio Renderer	Internal	Render DirectShow SDK library	Create() Init() Start() Terminate()
TSDemux library	TSDemux lib	External	Provide library function to perform TS Demuxing	TSDemuxInitialize() TSMain() PESInitialize() PESMain()
AVC DLL	AVC DLL	External	Provide library functions to decode MPEG-4 AVC AUs	Jm_init() jm_decode()
BSAC DLL	BSAC DLL	External	Provide library functions to decode MPEG-4 BSAC AUs	Init_BSAC() Decoding()
Image Library	CxImage lib	External	Provide library functions to decode and render JPEG images	Refer to the CxImage reference
ISO library	ISO lib	External	ISO library is used to parsing the mp4 file format and boxes of DMB-AF files.	Refer to the ISO Base Media File Format reference
DirectShow SDK library	DirectShow	External	DirectShow SDK library is used to render the audio and video by finding the appropriate filter to render the audio and video PUs.	Refer to the DirectShow reference

## 9.5 Metadata Browser

### 9.5.1 Introduction

The DMB-AF player plays the DMB-AF file conforming to the 'dv2a' brand and 'dv3a' brand with 3GPP Timed Text and TV Anytime (TVA) metadata. It renders the audio and video as a movie (the video is synchronized with the audio), with timed text embedded within the screen or in the text box, and the corresponding TVA metadata that can be used to browse the DMB-AF content, jump to segment of the video, and show the usage history (as for the current version of the RSW, only already existing usage history can be shown, which means the player is not recording the user action when users perform any action to the content in the player).

### 9.5.2 Software architecture

The DMB-AF player has the functional modules as shown in Figure 21. The description for each module is described in Table 13.

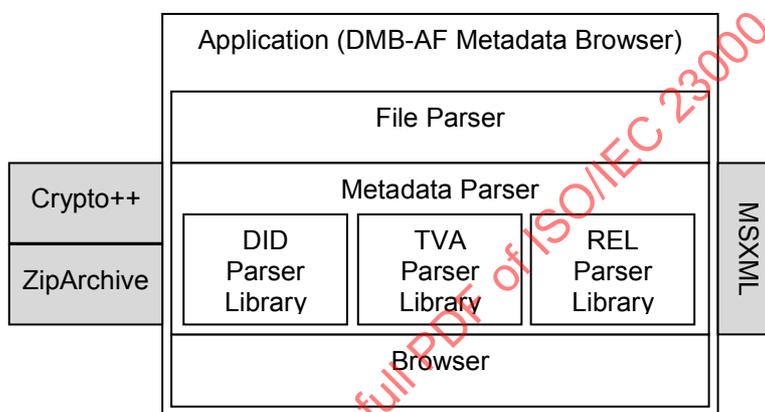


Figure 21 — Functional modules of Metadata Browser

Table 13 — List of functional modules in Metadata Browser

No.	Function	Type	Description
1	File parser	Internal	Modules related to load the DMB file, extract the metadata and parse the metadata
2	Browser	Internal	Displaying the metadata
3	DID Parser Library	Internal	Parse the DID and IPMP metadata
4	TVA Parser Library	Internal	Parse the TVA metadata
5	REL Parser Library	Internal	Parse the REL metadata
6	MSXML	External	Provides DOM API for parsing XML
7	Crypto++	External	Provides algorithm for decrypting protected contents
8	ZipArchive	External	Provides function to unzip Java MIDlet

### 9.5.3 Functional description

The DMB-AF metadata browser has the software architecture as shown in Figure 22. The description of each module is described in Table 14.

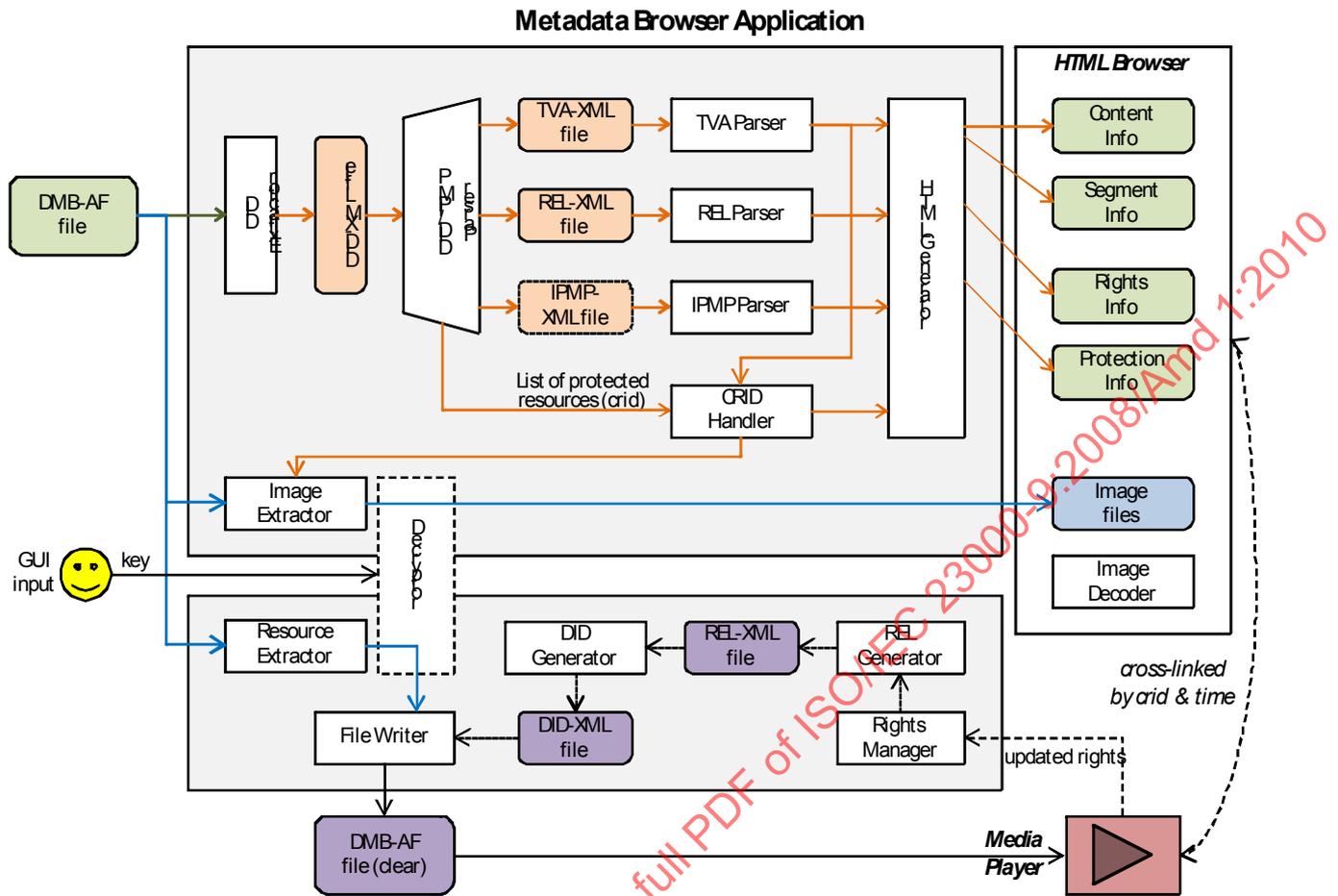


Figure 22 — Software architecture of Metadata Browser

Table 14 — List of implementation modules in Metadata Browser

Module	Functionality	Description	Implementation
File parser	File parser	Loads the DMB-AF file and extract the DID metadata inside the META box	OnOpen() ExtractDMB() ExtractREL ExtractTVAContent() ExtractTVASegment()
DID Parser	DID Parser	Loads the DID XML data and parse the DIDL and IPMP metadata structure	getIPMPGeneralInformation() getIPMPDIDLInformation
TVA Parser	TVA Parser	Loads the TVA XML data and parse the metadata structure	TVAGetSegmentGroupInformation () TVAGetSegmentGroupListIndex () TVAGetSegmentListIndex () TVAGetSegmentInformation () TVAGetComponentIndex() TVAGetItems () TVAGetContentDescription() isTVA()
REL Parser	REL Parser	Loads the REL XML data and parse the metadata structure	RELGetGrantLicense() isREL()
Browser	Mini browser	Displays the TVA metadata in form of HTML using Internet Explorer library. The event in the mini browser can also be used to control the video rendering (in segment browser application).	CreateHTML() Navigate()
MSXML2	MSXML2	MSXML2 is a library for building XML-based application.	Refer to the reference of the library
Crypto++	Crypto++	Provides algorithm to decrypt protected contents	Refer to the reference of the library
ZipArchive	ZipArchive	Provides function to unzip Java MIDlet contents	Refer to the reference of the library

## 9.6 Handling NPAD components

### 9.6.1 Introduction

DMB-AF can store NPAD (Non-Program-Associated Data) components such as BWS (Broadcast Website) and Java MIDlet contents.

### 9.6.2 BWS packaging and browsing

#### 9.6.2.1 Packaging BWS

To package BWS contents into DMB-AF packager, the BWS contents shall be already contained in a single directory that links to each other in the main web page file (usually named 'index.html').

Adding BWS package into DMB-AF is the same as adding TVA resources into DMB-AF. However, the information about the relative path of each resource of BWS contents is stored in the item information ('iinf') box as well as the crid and item location information. Figure 23 illustrates the BWS package (a package means a directory containing complete web page) and how to store the package into DMB-AF file.

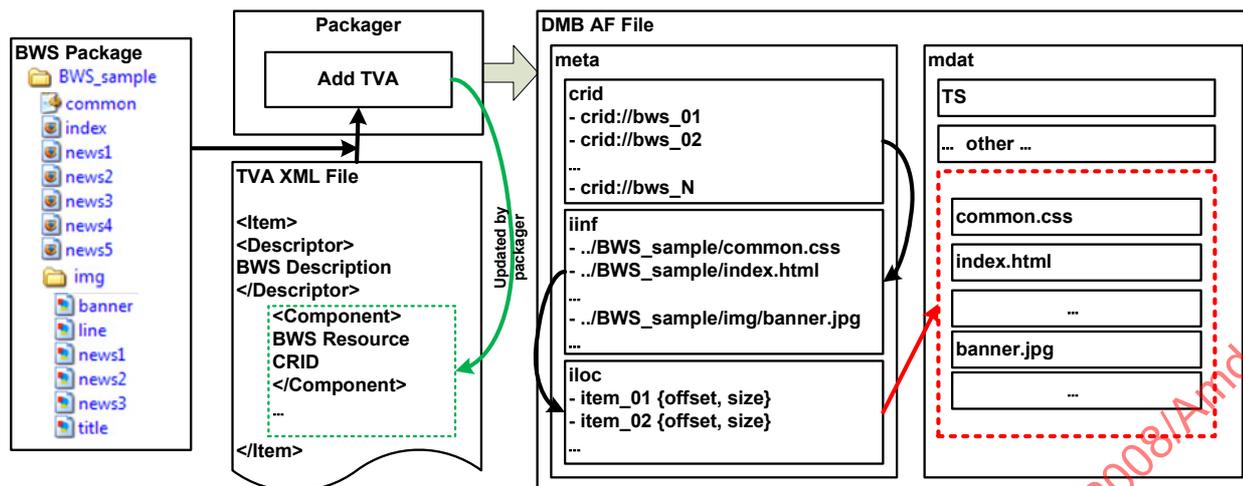


Figure 23 — Adding BWS into DMB-AF file

### 9.6.2.2 Extracting and browsing BWS

To extract and display BWS page from DMB-AF file, the extractor first parses TVA to obtain crid information of BWS. Based on the crid, the browser reads the corresponding item information and item location boxes. The description in item information is used to define the linkage structure of BWS contents. Therefore, when the contents are extracted from DMB-AF file based on the item location information, the player can regenerate the web page linkage structure of the BWS. Figure 24 illustrates the concept.

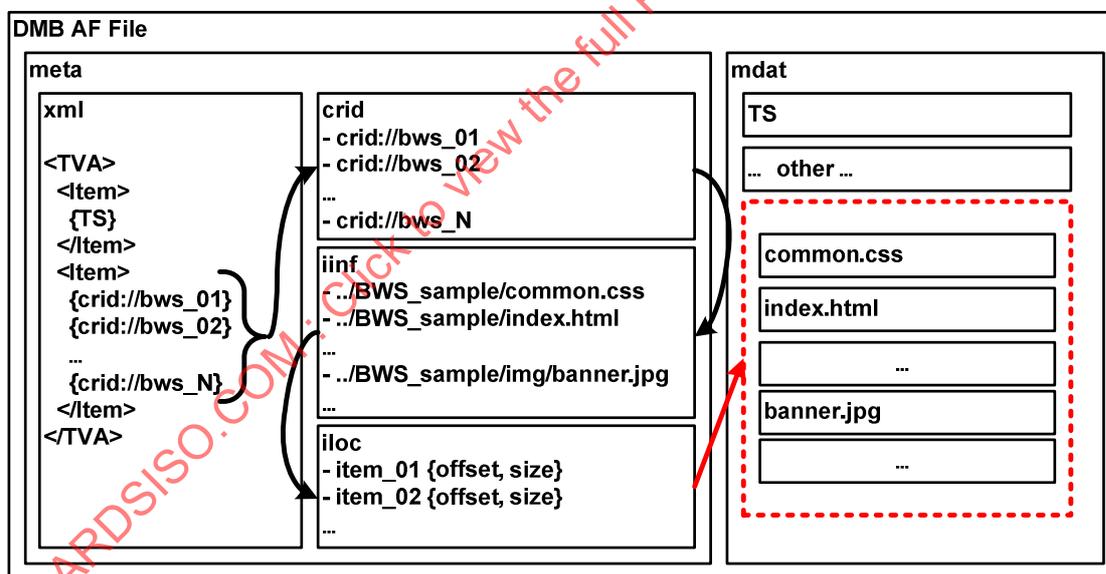


Figure 24 — Extracting BWS from DMB-AF file

### 9.6.3 JAVA MIDlet packaging and browsing

#### 9.6.3.1 Packaging MIDlet application

To package Java MIDlet contents into DMB-AF packager, the Java MIDlet contents shall be already be compressed in a zip file.

Adding Java MIDlet package into DMB-AF is the same as adding TVA resources into DMB-AF. However, in packaging the AF file, the information about the relative path of Java MIDlet package is stored in the item information ('iinf') box as well as the crid and item location information. Figure 25 illustrates the Java MIDlet package (a zip file) and how to store the package into DMB-AF file.

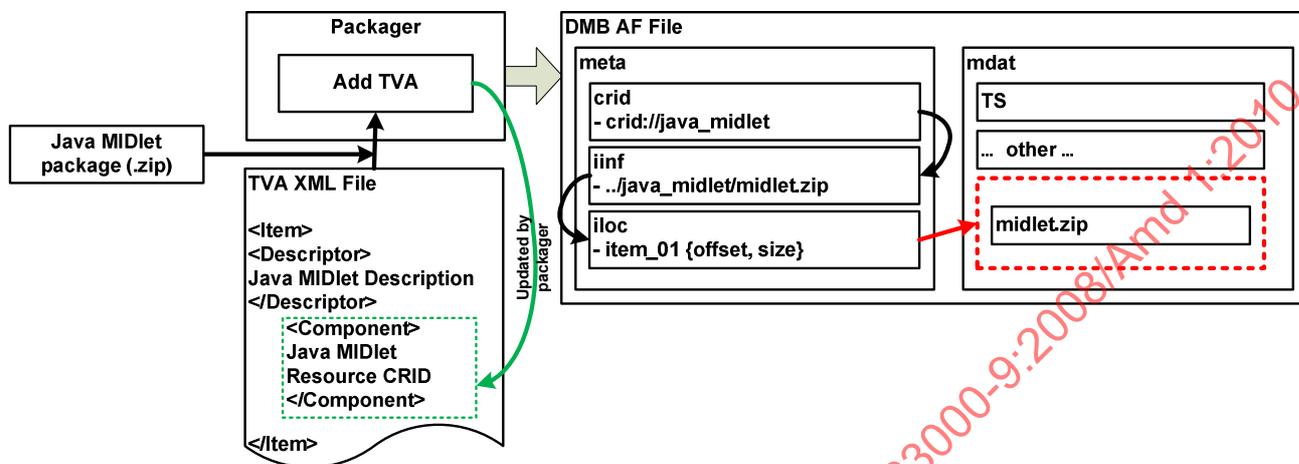


Figure 25 — Adding Java MIDlet into DMB-AF file

### 9.6.3.2 Extracting and browsing MIDlet application

To extract Java MIDlet page from DMB-AF file, the extractor parses TVA to obtain crid information of Java MIDlet. Based on the crid, the browser reads the corresponding item information and item location boxes. The description in item information is used to define the linkage of Java MIDlet package. When the contents are extracted from DMB-AF file based on the item location information, the browser then unzip the Java MIDlet package into temporary directory. Finally, the browser calls the MIDlet executor to run the unzipped Java MIDlet package. Figure 26 illustrates the concept.

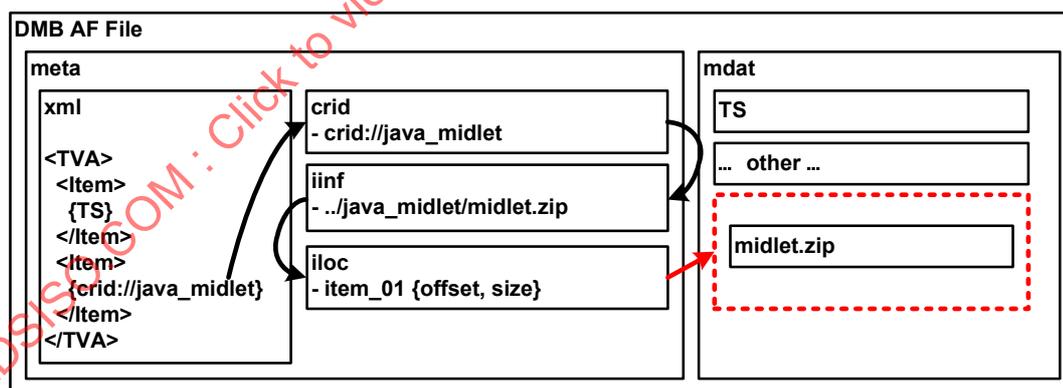


Figure 26 — Extracting MIDlet from DMB-AF file

Page 44, before the Bibliography

Insert the following two new annexes.

## Annex D (informative)

### Installation of reference software

#### D.1 Installation

This reference software is build using Microsoft Visual C++ from Microsoft Visual C++ 6.0 with Service Pack 6.0 in Windows XP SP2 and SP3 platform. The reference software has not been tested in any other development tools and/or platform (e.g. Microsoft Visual Studio 2008 on Windows Vista).

Follow these steps to make the reference software work appropriately in your station.

1. Download JDK (Java Development Kit) and JMF (Java Multimedia Framework) from the following websites and install them:
  - a. JDK: [http://java.sun.com/javase/downloads/index\\_jdk5.jsp](http://java.sun.com/javase/downloads/index_jdk5.jsp) (jdk-1\_5\_0\_21-windows-i586-p.exe)
  - b. JMF: <http://java.sun.com/javase/technologies/desktop/media/jmf/2.1.1/download.html> (jmf-2\_1\_1e-windows-i586.exe)
  - c. Note that the JDK install default path is "C:\Program Files\Java\jdk1.5.0\_21". If you install the JDK on a different path, you shall register an environment variable of "JAVA\_HOME" with the path using the "System Property" menu of Windows XP.
2. Extract the reference software zip file into a desired directory. The directories will be created as described in 9.2.
3. Open each of the following project workspace files:
  - a. proj/1\_Packager/1\_Packager.dsw
  - b. proj/2\_Media Player/2\_Media Player.dsw
  - c. proj/3\_Metadata Browser/3\_Metadata Browser.dsw
4. Install Microsoft DirectX 9.0b SDK (release date: July 24<sup>th</sup> 2003) and set the include and library paths using "Tools/Options/Directory" menu of the Microsoft Visual C++.
5. Rebuild all and run each application.

#### D.2 Known issues and limitations

The Packager can only be used to package well-prepared DMB-compliant TS file or MP4 file, TVA XML file, REL XML file, BWS file package (a web file structure), zipped JAVA MIDlet package, and plain text file with synchronization information using "/". It cannot be used to edit or modify existing DMB-AF file, it cannot be used to create the bitstreams and it does not check the conformance of input bitstreams. But it can add metadata to the existing DMB-AF files that contain only media data but metadata.

## Annex E (informative)

### User guide for reference software

#### E.1 Packager

##### E.1.1 Introduction

The DMB-AF Packager application provides the user interface to add various DMB contents and metadata files to be packaged into ISO-Based file format compliant to the DMB-AF specification. This annex briefly describes the step to create some of DMB-AF conformance files as described in Table 8.

Figure E.1 shows the user interface of the Packager.

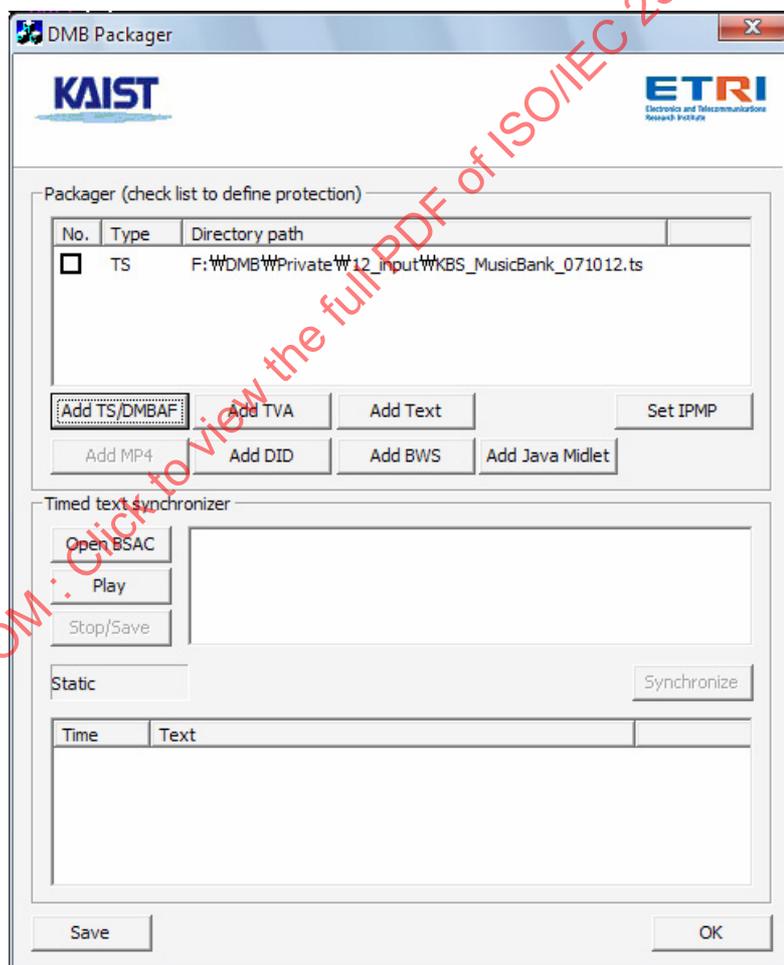


Figure E.1 — User interface of the packager

### E.1.2 Creating conformance file 01

1. Prepare one TS file conforming to T-DMB Profile 1, TVA metadata file, REL metadata file, supporting JPEG files for TVA item and segment information and timed text input file.

The timed text input file can be created by adding separator "/" in appropriate positions to the original text. The words separated by "/" becomes a basic synchronization unit which will be displayed at the same time. For example, the following lyric text:

"Good morning to you, good morning to you."

might be formatted into:

"Good /mor/ning /to /you, /good morning to /you /" (ends with "/")

2. Press "Add TS/DMBAF" button to add TS file.
3. Press "Add TVA" button to add TVA metadata file. A new window will appear to let you add supporting JPEG files for TVA items and segments, which content identification (CRID) are shown in the list. To add the JPEG files press "Open dir." button to select a directory that contains all JPEG files that have the same name as the CRID. Finally select the resource you want to add by selecting the rows in the list until the check boxes are checked. The JPEG image will be shown in the right side of the window. Press OK to continue.
4. Press "Set IPMP" to add REL metadata file. A new window will appear to let you add REL file containing pre-defined license information. Check the "Use REL file" selection and add the file. Left the other input values blank then press "OK" to continue.
5. To synchronize the timed text with audio, if you already add the Text file into the packager, firstly select any audio files encoded with BSAC specification using "Open BSAC" button. "Play" the audio and listen to the audio, the timestamp will be shown as time information for synchronizing the timed text. Press "Synchronize" button during the audio playing to synchronize the audio and the text. The synchronized text and its timestamp information will be added in the List box at the bottom.
6. Finally press "Save" to save DMB-AF file.

### E.1.3 Creating conformance file 02

1. Prepare a file with separated tracks, TVA metadata file, REL metadata file, and supporting JPEG files for TVA item and segment information.
2. Follow the steps 2 ~ 4 in E.1.1
3. Finally press "Save" to save DMB-AF file.

### E.1.4 Creating conformance file 03 or 04

1. Prepare one TS file conforms to T-DMB Profile 1 for conformance file 3 or one ES file with separated tracks for conformance file 4
2. Press "Add TS/DMBAF" button to add TS or ES file
3. Finally press "Save" to save DMB-AF file.

### E.1.5 Creating conformance file 05

1. Prepare one TS file conforms to T-DMB Profile 1, TVA metadata file, REL metadata file, and supporting JPEG files for TVA item and segment information