INTERNATIONAL STANDARD

ISO/IEC 14496-5

Second edition
2001-12-15

AMENDMENT 4
2004-08-15

Information technology — Coding of audio-visual objects —

Part 5: Reference software

AMENDMENT 4: IPMPX reference software extensions

Technologies de l'information — Codage des objets audiovisuels —

Partie 5: Logiciel de référence

AMENDEMENT 4: Extensions du logiciel de référence IPMPX

Reference number
ISO/IEC 14496-5:2001/Amd.4:2004(E)

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
Web 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 4 to ISO/IEC 14496-5:2001 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

Information technology — Coding of audio-visual objects —

Part 5:
Reference software

AMENDMENT 4: IPMPX reference software extensions

1 Scope

Add the following lines to Clause 5:

IM1/IPMPNull
IM1/IPMP
IM1/Lib
IM1/BinaryMessagesLib
IM1/IPMP_DESTool
IM1/IPMP_DummyTool
IM1/IPMP_MasterTool
IM1/IPMP_RELTool
IM1/IPMPXFull

IPMP-H (Hooks) sources

IPMP-X (eXtensions) sources

Add the following lines to Annex C (Providers of reference software):

Panasonic Singapore Laboratories (PSL)

Central Research Laboratories Ltd (CRL)

Multimedia Architectures

Add the following paragraphs (till the end of this document) just before Annex C (Providers of reference software) and rename the existing Annex C as Annex D:

Annex C
(informative)

Guidance of IPMP software

C.1 Introduction

C.1.1 How to get the codes

This annex identifies, describes and explains the provided reference software that implements the MPEG-4 IPMP Extension specifications found in ISO/IEC 14496-1 and ISO/IEC 14496-13.

The reference software, which is based on IM1, includes all the normative parts of the MPEG-4 IPMP Extension standard, and some informative part as well. The reference software can be retrieved from MPEG CVS server under path "MPEG-4/System2".

C.1.2 How to compile

To retrieve the reference software from MPEG CVS server, open WinCVS, click on the parent directory of MPEG-4/ directory. Then, go to menu Remote/Checkout Module and in the Checkout settings dialog, under "Module name and path on the server", enter MPEG-4/System2 (warning: this is case sensitive!).

Please go to MPEG-4/Systems2/IM1/Docs/ and read _ReadmeFirst.doc which is necessary for you compile IM1. Follow all the steps, one after another. This will help you configure your machine correctly, download whatever missing software required by players in order to compile BifsEnc, Mp4Enc, Im1Player, IM1-2D, Osmose, Player3D with MPEG-J support.

As for IPMP-X portion, a simple **user notes document** can be found in IPMP-X directory under IM1 directory to provide the initial guidance.

C.1.3 Source codes directory

Please follow the user notes document under IPMP-X directory when you operate on the source codes.

IM1/IPMP-H contains most of IPMP Hooks related files, including the following three sub-directories.

- /IPMPNull
- /IPMP
- /Lib

IM1/IPMP-X contains most of IPMPX related files, including the following sub-directories.

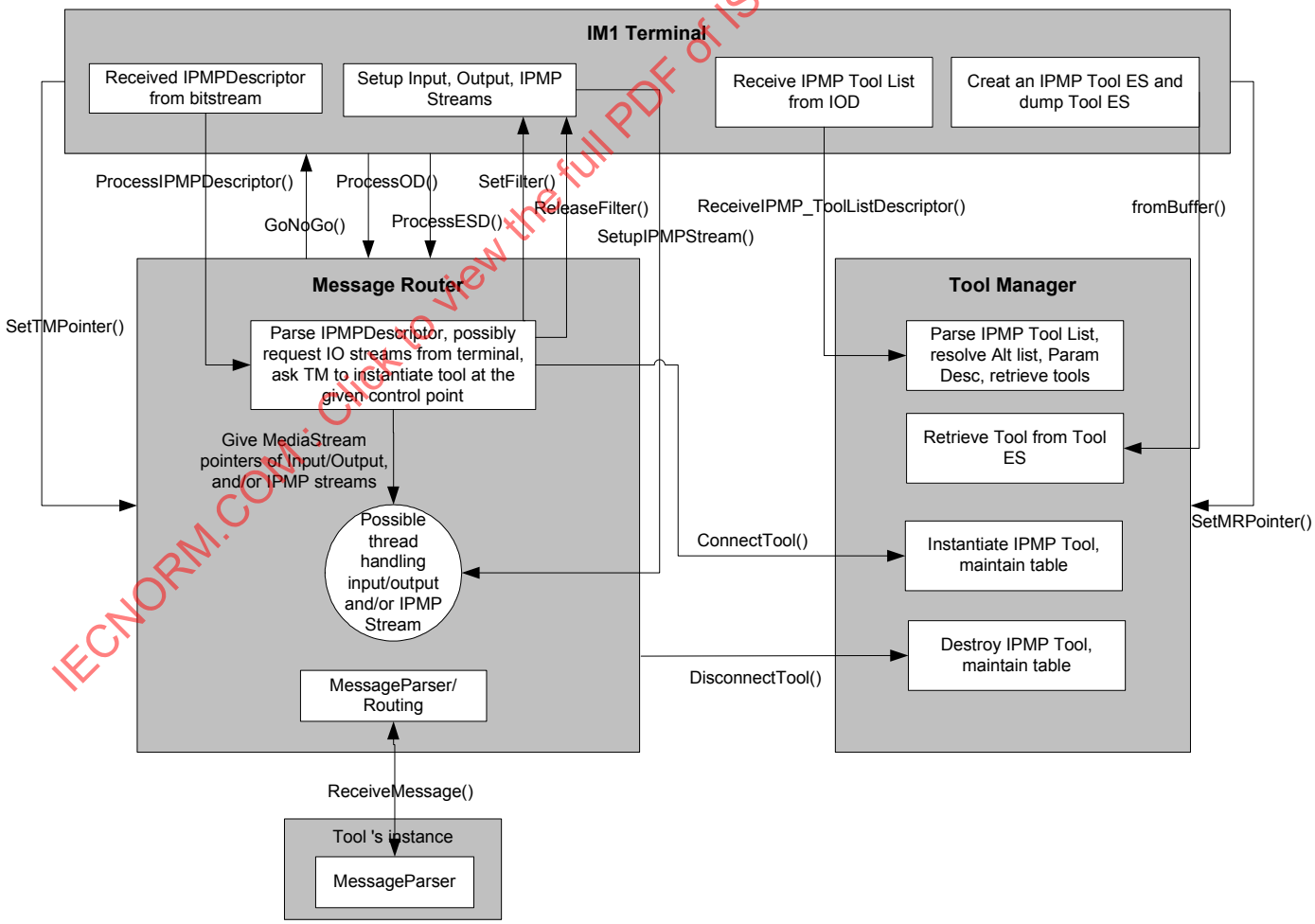
- /BinaryMessagesLib MPEG-4 messages libraries and messages bitstreams
- /Include:
- /IPMP_DESTool DES decryption tool
- /IPMP_DummyTool Dummy tool
- /IPMP_MasterTool Master tool, working with REL tool
- /IPMP_RELTool REL tool
- /IPMPXFull IPMPX Framework

- /Lib
- /TestFiles
  - t2.723 audio stream file (used by all test cases)
  - t1.263 video stream file (used by all test cases)
  - Six sets of testing files (.txt, .scr .mp4 and .trif), including *simple* (non-ipmp), *ipmpx_REL* (with REL tool), *ipmpx_Dummy* (with dummy tool), *ipmpx_REL-Dummy* (with both dummy tool and REL tool), *ipmpx_Dummy(TES)* (with dummy tool carried in IPMP Tool ES), *ipmpx_IPMPS_DES* (with DES tool working with IPMP Stream)
  - Some license files and user query files in .xml format, at least including conditionBothLicense.xml, exampleInput2.xml, exampleInput2.xml and exampleInput_illegal.xml (used by *ipmpx_REL* and *ipmpx_REL-Dummy*)
  - DESDLL.dll for DES Encryption & Decryption (used by *ipmpx_IPMPS_DES*)
  - DESKey_IPMPS.bin IPMP Stream file (used by *ipmpx_IPMPS_DES*)
  - DESKeysFile.txt containing two hundred 128 bit keys (used by *ipmpx_IPMPS_DES*)
  - ToolAU-Dummy.bin IPMP Tool Elementary Stream file (used by *ipmpx_Dummy(TES)*)

The IPMP.dsw under IM1/_WorkSpaces associates to IPMP related codes. Please use that to open IM1-core projects and IPMP related projects.

C.2 Details of MPEG-4 IPMP Extension Reference Software

C.2.1 Architecture



C.2.2 Interfaces among IM1 Terminal, Message Router, IPMP Tool Manager, IPMP Tool:

MPEG-4 IPMPX standardizes a set of IPMP messages, but not the interfaces that pass the messages. The exact interfaces depend on the actual industrial domain. The interfaces that are implemented in the reference software are not normative.

IM1 Terminal:

```
// GoNoGo is called by the MR (Message Router) with a true if processing the stream
    // may begin and a false if processing must stop.
    bool GoNoGo(bool go);

    // Called by the MR to create a control point.
    bool SetFilter( Filter* filter, int controlPoint );

    // Called by the MR to release a control point.
    bool ReleaseFilter( Filter* filter );
```

Message Router (implemented in IPMPXFull)

```
    // setting TM (Tool Manager)'s pointer, so that MR can use this to access TM's member functions
    bool SetTMPointer(TM_Interface * tmPointer);

    // Called by the Terminal to process an IPMPDescriptor.
    bool ProcessIPMPDescriptor(IPMP_Descriptor* descriptor);

    // Called by the Terminal to process an Object Descriptor.
    bool ProcessObjectDescriptor( ObjectDescriptor* pOD);

    // Called by the Terminal to process an Elementary Stream Descriptor.
    bool ProcessESDescriptor( ESDescriptor* pESD);

    // Called by the Terminal to remove an IPMPDescriptor.
    bool RemoveIPMPDescriptor(IPMP_Descriptor* descriptor);

    // Called by the Terminal to notify MR that an Object Descriptor was removed
    bool RemoveObjectDescriptor( ObjectDescriptor* pOD);

    // Called by the Terminal to notify MR that an Elementary Stream Descriptor was removed
    bool RemoveESDescriptor( ESDescriptor* pESD);

    // Called by the Terminal to create a sink for an IPMP stream.
    bool SetUpIPMPStream(ObjectDescriptor *pOD, ES_Descriptor* pESD, MediaStream* ipmpStream);

    //Called by the Terminal to destroy a sink for an IPMP stream that was previously created by the Terminal
    bool RemoveIPMPStream(ES_Descriptor *pESD);

    // messaging interface
    bool ReceiveMessage(IPMP_ToolMessageBase* msg);
```

IPMP Tool Manager (implemented in IPMPXFull)

```
    // setting message router's pointer, so that TM can use this to access MR's member functions
    bool SetMRPointer(MR_Interface * tmPointer);

    // Called by the MR to indicate a tool is no longer needed.
    bool DisconnectTool(void* toolPtr );

    // Called by the MR to request a tool be instantiated.
    void* ConnectTool(ES_Descriptor *pESD, IPMP_Descriptor* descriptor);
```