INTERNATIONAL STANDARD

ISO/IEC 23004-8

First edition
2009-04-01

Information technology — Multimedia Middleware —

Part 8: Reference software

Technologies de l'information — Intergiciel multimédia —

Partie 8: Logiciel de référence

Reference number
ISO/IEC 23004-8:2009(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

Contents

Page

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.

ISO/IEC 23004-8 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

ISO/IEC 23004 consists of the following parts, under the general title *Information technology — Multimedia Middleware*:

— *Part 1: Architecture*

— *Part 2: Multimedia application programming interface API*

— *Part 3: Component model*

— *Part 4: Resource and quality management*

— *Part 5: Component download*

— *Part 6: Fault management*

— *Part 7: System integrity management*

— *Part 8: Reference software*

Introduction

ISO/IEC JTC 1/ SC 29 has produced many important International Standards (for example MPEG-1, MPEG-2, MPEG-4, MPEG-7, and MPEG-21). One of the next steps in this process is the standardization of an Application Programming Interface (API) for Multimedia Middleware (M3W) allowing application software to execute multimedia functions with a minimum knowledge of the inner workings of the multimedia middleware as well as to support a structured way of updating, upgrading and/or extending the multimedia middleware.

Information technology — Multimedia Middleware —

Part 8:
Reference software

1 Scope

This part of ISO/IEC 23004 explains the organization of the reference software for ISO/IEC 23004– 1 to 7 (Multimedia Middleware). The electronic attachment to this part of ISO/IEC 23004 provides the source code of the actual software.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 23004-2, *Information technology — Multimedia Middleware — Part 2: Multimedia application programming interface (API)*

ISO/IEC 23004-3, *Information technology — Multimedia Middleware — Part 3: Component model*

ISO/IEC 23004-5, *Information technology — Multimedia Middleware — Part 5: Component download*

ISO/IEC 23004-6, *Information technology — Multimedia Middleware — Part 6: Fault management*

3 Overview of reference software

This is an informative clause. The reference software is organized into directories according to the different parts of ISO/IEC 23004. These directories are:

— **1_Architecture:** This directory is rather empty. The architecture is reflected by the implementation of the other parts.

— **2_Multimedia-API:** This directory contains the reference implementation of Audio and Video, Governance and IPMP logical components. The Audio and Video logical components are based on UHAPI4Linux implementation.

— **3_ComponentModel:** This directory contains the implementation of the core framework, services for remote method invocation (REMI) and services that allow instantiation of services based on a logical component id (Service Manager). The core framework also contains tools that aid in the development of M3W Components (IDL compiler).

— **4_ResourceManagement-Framework:** This directory contains the implementation of the resource management framework. This framework can be used to optimize the Quality of Service perceived by the user in a situation where resources are constrained and often not enough to run all applications and services at the highest quality level.

— **5_ComponentDownload-Framework:** This directory contains the implementation of the download framework. A framework that enables a large number of scenarios of controlled download / upload of components

— **6_FaultManagement-Framework:** This directory contains the implementation of the fault management framework enabling transparent addition of fault tolerance techniques to your software. The implementation consists of policies for intercepting the creation of services and tools for generating wrappers (middleman) that contain the fault tolerance techniques.

— **7_IntegrityManagement-Framework:** This directory contains the implementation of the integrity management framework targeted at maintaining and restoring in consistent software configuration on a device in the period that a device is owned and used by a consumer.

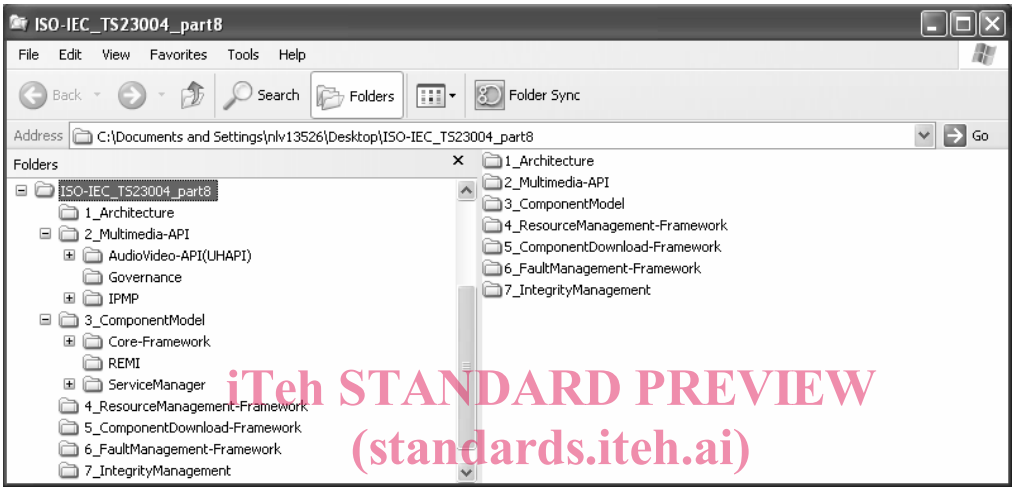

Figure 1 — Organization of the reference software in directories

4 Multimedia API

4.1 Introduction

This is an informative clause. This clause gives a brief explanation on the parts that together form the reference implementation for the Multimedia API of M3W.

4.2 Audio Video

4.2.1 General

The directory contains the current release of the reference implementation for clauses 7(general interfaces), 8 (audio interfaces) and 9 (video interfaces) of ISO/IEC-23004-2 (clause numbers are according to the FDIS version).

The implementation is based on the UHAPI4Linux implementation that can be found on Sourceforge (www.sourceforge.net).

4.2.2 Building

For building instructions please look at:

— `2_Multimedia-API\AudioVideo-API(UHAPI)\HOWTO`

4.2.3 Functionality

The reference implementation enables you to build a simple television application with features like changing channels, adjust volume, adjust contrast, adjust contrast, adjust saturation, image overlay, etc.

Figure 2 — Screenshot of Television Application build using reference implementation of Audio Video logical components

4.3 Governance

4.3.1 Introduction

The function of Security Manager is to provide secure interaction between M3W Service and external entities. This may include the interaction to acquire Service through download mechanism and invoke Service from remote peer. Security Manager also provides interface for governance/permission checking for accessing M3W Service and/or platform's resources.

4.3.2 Implementation

Currently the implementation consists of one executable component with a SecurityManager service that implements the ISecurityManager interface suite (logical component) and an example client application that uses this service.

Service and application are build in the following way:

- autoreconf –is
- ./configure –prefix=$HOME/local
- make
- make install

Please remember to register the component and its service after installation but before execution of the client application. This is done by the following command:

- rcregtool < regscript

The registration script "regscript" is generated during the build process.

4.4 IPMP

4.4.1 Introduction

ISO/IEC 23004-2 M3W multimedia API has been developed as a reference software. This clause describes the reference software that provides the M3W middleware interface for MPEG-2 IPMP [3], MPEG-4 IPMPX [4], and MPEG-21 IPMP [5] services. It will refer to IPMP interface for supporting IPMP service of a device part of M3W.

The implementation is M3W complaint except for:

— one operation name in the `rcIComponent (getServiceFactory())` and the interface name `rcIServiceFactory`. The current release still uses the legacy names `getServiceManager` and `rcIServiceManager`.

— The usage of a service specific interface instead of the `rcIServiceGeneric` interface

The IPMP interface suite is listed as following Table 1.

Table 1 — IPMP interface suite list

IPMP Interface Suites		
Name	Path	Functional Description
Trust Management		
Key Management	~/IPMP/component /trustmgmt/keymgmt	Key management component is essential for IPMP functions such as encryption, decryption, watermarking, and authentication and is used to process encrypted multimedia data in accordance with getting and generating key information.
Signature Management	~/IPMP/component /trustmgmt/signmgmt	Signature management component is important to represent immunity of IPMP properties. A digital signature means a value generated from the application of a private key to a message via a cryptographic algorithm such that it has the properties of integrity, message authentication and/or signer authentication.
License Management	~/IPMP/component /trustmgmt/licensemgmt	In DRM system, usage rule and copyright are written in a license with key information. License management component is a representation of right expression for digital contents.
Certificate Management	~/IPMP/component /trustmgmt/certmgmt	Certificate management component can be used in managing key and related information.
Domain Management	~/IPMP//component /trustmgmt/domainmgr	Domains management component are supporting groups of devices or users to share some common properties. These are defined to manipulate domain elements such as domain information, device information and so on.

Tool Interfaces		
General Tool Processing	~/IPMP/component /tool/toolproc	According to tool processing component, the required IPMP tools can be selected and initialized. After that, the multimedia data are processed by tool processing component and the selected tool dose its duties based on IPMP messages.
IPMP Tool Functions	~/IPMP/component /tool/toolfunc	Any kind of functions can be supported in a device, if the functions are related to IPMP tool functions component. However, IPMP tool functions component defines a tool function of general functionalities for DRM or IPMP services such as watermark and authentication functions.
Tool Update	~/IPMP/component /tool/toolupdate	Tool update component should support updating and installing IPMP tool and upgrading the functionality of the tool.
Tool Communications	~/IPMP/component /tool/toolcom	Communications between different tools or IPMP devices should be supported by tool communication component.

Table 1 contains path name and functional description with implemented component name. And all components are implemented by C/C++ language.

We have classified components with *trust management* interfaces and *tool* interfaces. The trust management interfaces are to support tool's functionality by accessing secured information. The tool interfaces are to perform a variety of IPMP functions based on IPMP information. Also, components are developed in IDL (Interface Description Language) and IDL type's components have been compiled by M3W IDL compiler. After that, IDL type's components will be C/C++ type components. For more details of IDL compiler usage, we recommend referring to ISO/IEC 23004-3.

The RIDL type of component needs common type definition for IPMP interface suite. Therefore, we have defined types in a file "ipmp_types.ridl." When "ipmp_types.ridl" is compiled by IDL compiler, we can get two files; "ipmp_types_common.c","ipmp_types_common.h." These two files are required whenever our proposed component is registered in M3W runtime environment.

4.4.2 Example of IPMP Interface Suites

We have implement examples for verifying components for IPMP interface suite whether it is right to operate on M3W. A set of IPMP interface suite example is listed in Table 2. Note that the following examples are just calling interface functions of a component on M3W environment and their components are not actual operation for IPMP service.

Table 2 — Example of IPMP interface suites

Example		Path
1	IPMP processing	~/IPMP/example/IPMPprocessing
2	License management	~/IPMP/example/Licensemangement
3	Certificate management	~/IPMP/example/Certmanagement
4	Authenticating using a certificate	~/IPMP/example/Authentication
5	Add new domain	~/IPMP/example/AddNewDomain
6	Revoke domain	~/IPMP/example/RevokeDomain
7	Update domain	~/IPMP/example/UpdateDomain
8	IPMP Processing extension	~/IPMP/example/IPMPprocessingExt

Also, we can find out execution files for examples in the above path when archived file is installed in M3W environment.

The following is the description for IPMP interface suites example. Although examples do not have the functional process for IPMP service, they effectively show how to call an interface function of IPMP service component.

4.4.2.1 IPMP processing

IPMP terminal has all tools and do not need to find the missing tools. IPMP tool manager of IPMP terminal requests M3W's service manager to get the service instance of tools. And IPMP tool manager requests each tools to start `operateTool()`. This could be done by using `Start()` method in `RcIService` interface that is implemented by the IPMP tool service.

4.4.2.2 License management

License management component is how to check and get a valid license through license management component. When a tool requires license information, license management component should get and parse a valid license. After that it should return a result into the tool.

4.4.2.3 Certificate management

To certificate, we need operations such as issuing, updating, revoking for certificating a user with a certificate management component of M3W environment.

4.4.2.4 Authenticating using a certificate

When a device or a component has its certificate, it should be ready for authentication. If a device *A* wants to authenticate a device *B*, the device *A* requests the device *B*'s certificate and should validate it.

4.4.2.5 Add new domain

Domain Manager (DM) requests authentication of domain information such as administrator to Authenticator. Once Authenticator receives the required information from DM, Authenticator verifies the user which will be the domain administrator. If the information is valid, DM creates new domain and returns the result to Initiator.

