INTERNATIONAL STANDARD

ISO/IEC 23000-7

First edition
2008-08-15

AMENDMENT 1
2009-04-01

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

Part 7: Open access application format

AMENDMENT 1: Conformance and reference software for open access application format

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

Partie 7: Format pour application d'accès ouvert

AMENDEMENT 1: Conformité et logiciel de référence pour format d'application d'accès ouvert

Reference number
ISO/IEC 23000-7:2008/Amd.1:2009(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

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 23000-7:2008 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 — Multimedia application format (MPEG-A) —

Part 7:
Open acces application format

AMENDMENT 1: Conformance and reference software for open access application format

After Clause 6, add the following two new clauses:

7 Conformance

7.1 Introduction

This clause defines the conformance of implementations to the open access application format specified in the previous clauses.

7.2 File conformance

Conformant files shall be readable by the open access application format compliant implementations including the reference software as described in Clause 8. The general file-level structure of the files shall conform to the normative file structure defined in 6.8.

7.3 Player application conformance

A player application for content consumption is conformant to this standard, if the application can correctly read the file format, parse the Digital Item Declaration in the xml-box and display this information.

7.4 Creator application conformance

A file creator application is conformant, if the application can produce files conformant to the specification in Clause 6.

8 Reference Software

8.1 Introduction

The reference software can be used for creating and consuming conformant open access application format compliant files. The Java programming language is used for the implementation of this software. The application can be used to package arbitrary files as contents into a standard conformant file. The creation of these files is not part of the reference software.

8.2 Architecture

The following picture shows the high-level architecture of the open access application format reference software:

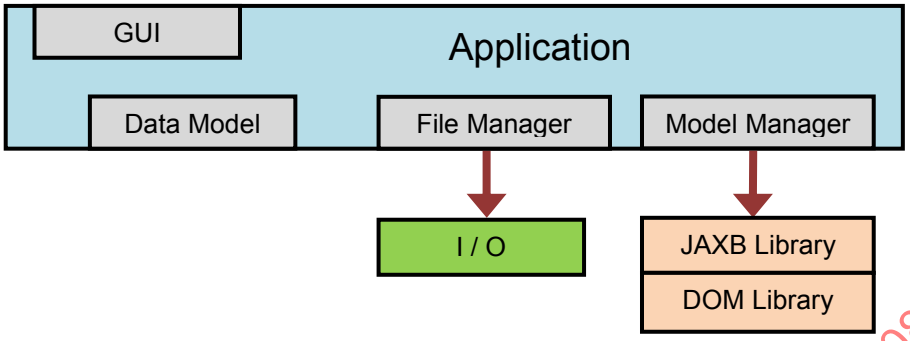

Figure 2 — Reference software architecture

The software is structured in the four main parts: the GUI, the Data Model, the File Manager and the Model Manager.

The GUI package contains the user interface for the interaction with the user. The inputs from the user are processed as events that call the respective functions in the Managers to process the Data Model.

The Data Model can be divided into the Content Model and the User Data Model. The Content Model is a representation of the data contained in the file format, which contains information about the resources and the metadata. The User Data model stores the information about the user of the application and also contains application preferences.

The Managers are divided in the File Manager and the Model Manager. The File Manager can be used to create, parse and save open access application format compliant files. These Open Access files are based on the MPEG-21 file format and shall be conformant to the specification in 6.8. Additionally the File Manager also provides functions to add, delete and save contents that are packaged within an Open Access file.

The Model Manager is specialized to process xml data and is used for the generation and parsing of XML using the JAXB and the DOM library. The Model Manager provides methods for the extraction of specific data from the model and the modification of the model.

8.3 Functional description

The reference software has two modes of operation: The content creation and the content consumption. The modes of operation are controlled in the GUI, which uses the functions of the Model Manager to process the data model for the creation and consumption of items. The functions of the File Manager are applied for the file creation and parsing and also for the resource handling.

8.3.1 Content creation

For the content creation the user can create a package file and add content files as items into the package. These content files can be arbitrary data. When the user finishes the packaging, an Open Access file can be generated, that can be released and consumed afterwards.

The user can specify the metadata for the item within the application. The information in the user interface is saved into the data model with the functions provided by the File and Model Managers. The main functions of these Managers are described in Table 2.

Table 2 — Functions for content creation

Function	Package and class name	Type	Description
addCreatorInformation()	jaxb.ContainerManager	Model Manager	Adds information about the creator of the package.
addItem()	jaxb.ItemManager	Model Manager	Inserts a new item into the package.
saveAuthors()	jaxb.ItemManager	Model Manager	Saves the information about the authors in the data model.
saveCopyrightString()	jaxb.ItemManager	Model Manager	Saves the string with the copyright information of the item.
saveCreationDate()	jaxb.ItemManager	Model Manager	Sets the date of the creation of the item in the data model.
saveEventReports()	jaxb.ItemManager	Model Manager	Attaches Event Report Requests to the item.
saveItemDescriptionPlain()	jaxb.ItemManager	Model Manager	Saves the plain text description of the item.
saveLicenseGrants()	jaxb.ItemManager	Model Manager	Stores the grants of the REL license in the data model.
saveLicenseURIs()	jaxb.ItemManager	Model Manager	Sets the URIs of the license in the data model.
saveLicenseURL()	jaxb.ItemManager	Model Manager	Saves the URL of the license of the item.
saveRelatedIdentifiers()	jaxb.ItemManager	Model Manager	Sets the relationships to other items.
saveTitle()	jaxb.ItemManager	Model Manager	Saves the title of the item.
releaseFile()	jaxb.ReleaseManager	Model Manager	Releases the current package and creates a compliant data model.
openFile()	file.FileManager	File handling	Opens a file and parses the boxes of the file format.
saveFile()	file.FileManager	File handling	Saves the contents in the Data Model in the boxes of the file format.
addResource()	file.FileManager	Resource handling	Opens a resource file and adds it to the package.
removeResource()	file.FileManager	Resource handling	Removes a resource file from the package.
extractResource()	file.FileManager	Resource handling	Extracts a resource file from the package and saves it as a new file.
NOTE The package names have the prefix "org.ldv.oa.manager." to provide a unique name for the integration in other implementations.			

8.3.2 Content consumption

For the content consumption the user can view and copy the content of an Open Access file. Furthermore the user can extract the content out of the file, which includes the transmission of Event Reports. The main functions of the reference software for content consumption are described in Table 3.

Table 3 — Functions for content consumption

Function	Package and class name	Type	Description
copyItem()	xml. XMLManager	Model Manager	Copies an item into another package.
validateItem()	dom. SigValidateManager	Model Manager	Validates the signature of an item if a signature is present.
sendEventReport()	jaxb.EventReportManager	Model Manager	Sends out the Event Reports if Event Report Requests are attached.
isAuthroized()	jaxb.AuthorisationManager	Model Manager	Validates if a right is authorized in the license.
extractResource()	file.FileManager	Resource handling	Extracts a resource file from the package and saves it as a new file.
NOTE The package names have the prefix "org.ldv.oa.manager." to provide a unique name for the integration in other implementations.			

8.4 Dependencies

The reference software needs an installed Java environment, which should be a Java JDK/JRE with version 1.5.0_12 or later.

Furthermore the following libraries are used in the software:

— Java Architecture for XML Binding, version 2.0

— JavaBeans Activation Framework, version 1.0.2

— Bouncy Castle Crypto APIs for Java, version 1.2.2

— JSR173: Streaming API for XML

— JavaMail API Design Specification, version 1.4.1

— Java XML Digital Signature, version 1.0.1

After Annex D, add the following new annex:

Annex E
(informative)

Reference software installation and user guide

E.1 Installing and compiling the source code

The reference software is contained in the attached zip.

The reference software contains an Apache ANT build file, which can be used to compile the software. The following steps are needed to install and compile the reference software:

1. Install Apache ANT (version 1.6.5 or newer) on the system.
2. Extract the contents of the zip file with the source code of the reference software in any directory.
3. Open the directory in a command shell. The sources are compiled by executing ant with the parameter "compile" in this directory. The software can be compiled and executed in one step using ant with the parameter "run".

E.2 User Guide

E.2.1 Initialization

On the first execution some user information is required for the execution of the software. The reference software uses two dialogs at the beginning to prompt the user for this information. The first dialog asks the user about his name and address, the second dialog requests the name of the Mail-Server, which is used to transmit Event Reports via e-mail.

E.2.2 File browsing and releasing

After the initialization the main window appears as shown in Figure E.1. The software works in a similar way as an archiving program.

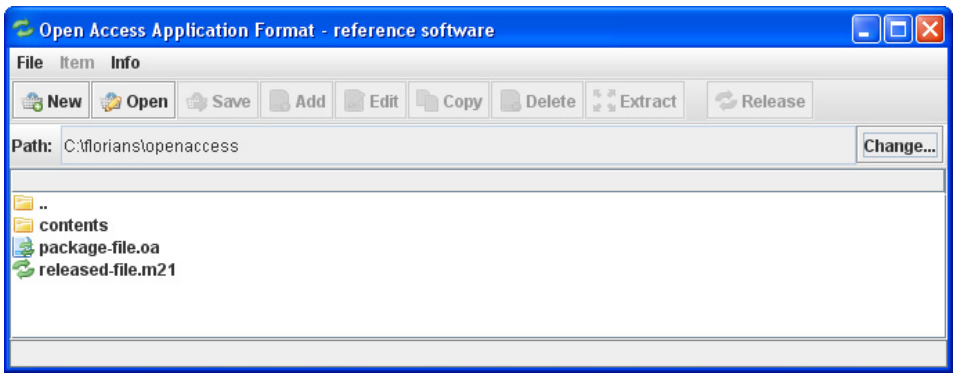

Figure E.1 — The main window

The program has a toolbar to create a new package file or to open an existing file. Below the toolbar the program shows a frame with the contents of the current directory. The user can browse through directories in this window and open files.

The software has two modes of operation: the content creation mode and the content consumption mode. When the user creates a new file or opens an existing package file, the program switches to the content creation mode. In this mode the frame changes and displays the contents of the package file. The user can modify the contents of the file and then save it with the extension "oa".

When the user finished modifying the package, he can choose to release the package. The software creates a standard compliant Open Access file, which can be saved with the extension "m21". This file can then be released and distributed. When the user opens a released file, the software switches to the content consumption mode.

E.2.3 Content creation

In this mode, the current list of items in the package file are displayed in the main window. The user can choose to add, edit, copy or delete items. To edit an item, the user can double click on an item in the list, which opens a dialog as shown in Figure E.2.

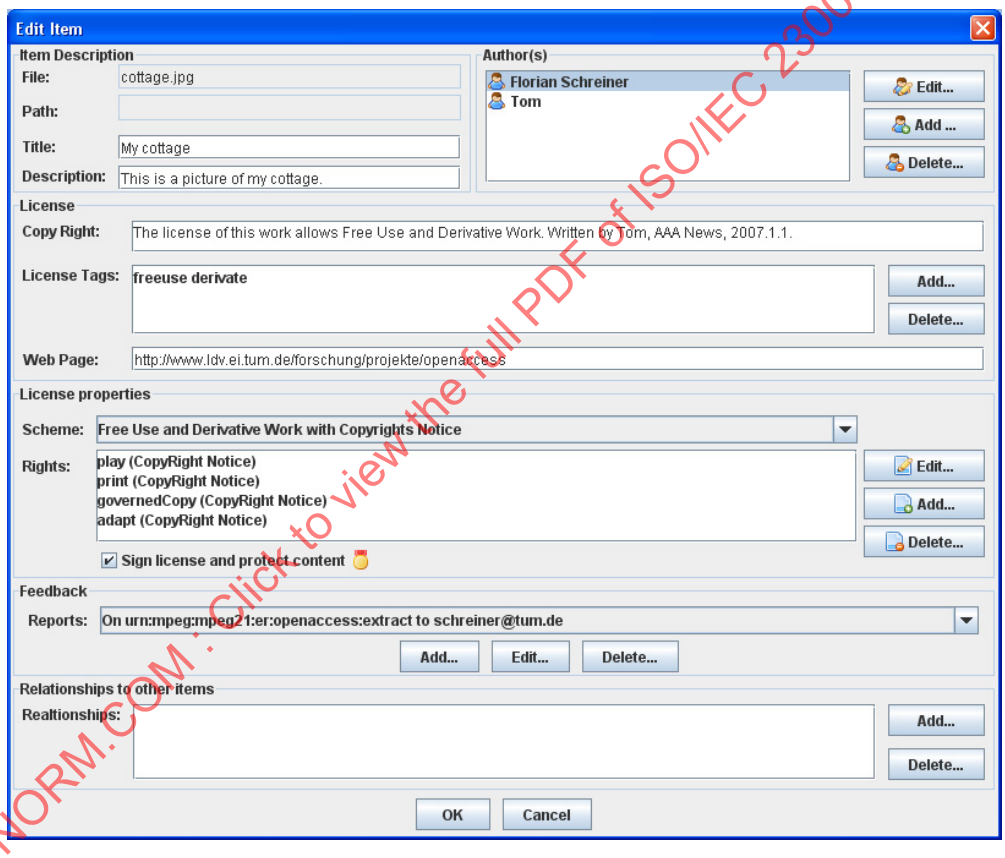

Figure E.2 — Metadata dialog of an item

The Figure gives an example for the metadata, which describes an image with the name "cottage.jpg". The user can provide information about the authors, the license, properties of the license, the feedback reports to the author and the relationships to other items. For each category dialogs appear, which ask the user for the necessary information.

E.2.3.1 Relationships between items

The section at the bottom of the dialog shown in Figure E.2 specifies the information about related items. The relationships are links to other items, which are either adaptations of the current item or the current item was derived from the related item. The user can add a new relationship by opening another Open Access file, which contains the related item. Then Figure E.3 appears, which shows the content of the other file. The user can choose the type of the relationship to the other item and then he can select the item in the list.

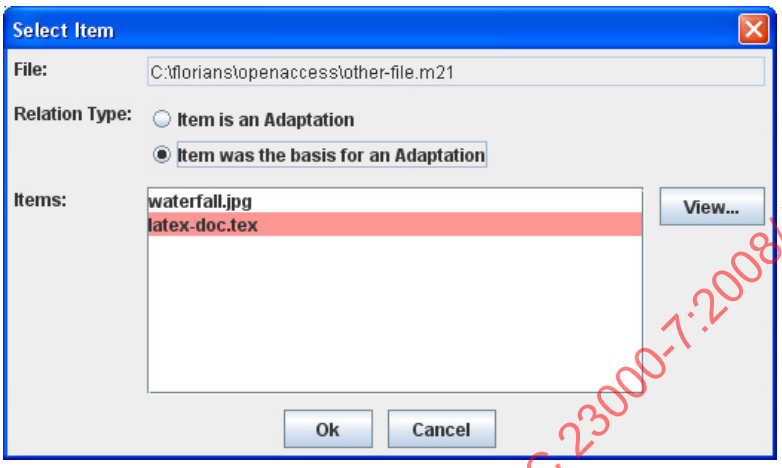

Figure E.3 — Dialog to choose the item to be copied

In this example the user wants to relate to another item, which he used to create the current item by adapting it, so the user chooses that the item was the basis for an adaptation. The program notifies the user by marking items with red colour if the item doesn't have the necessary license properties (Rights Expressions) to establish the relationship. In this example the image "waterfall.jpg" has the right for adaptation, while the document "latex-doc.tex" doesn't provide this right. After the user has selected an item, the relationship between the items is set.

E.2.3.2 Copying items

As shown in Figure E.1 the user can also copy items from other Open Access files into the package. Copying items means to select an item within another Open Access file and to insert it into the current package. To select an item from another file, a similar dialog as in Figure E.3 is shown. In this dialog items are marked in red if they don't provide a rights expression to copy the item. When the user has chosen the item, the license information of the item is shown. Then the item is inserted into the package as shown in Figure E.4 with the item "latex-doc.tex". The item is marked grey to show, that it has already been released in another file and that it cannot be changed any more in the current package.

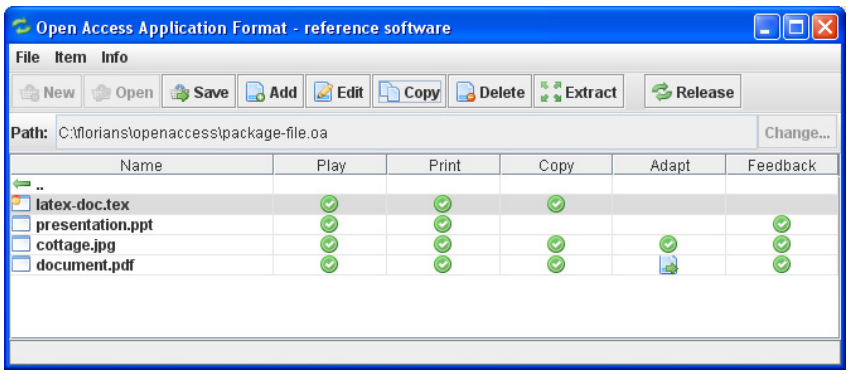

Figure E.4 — The main window with a copied item