Contents

1. SCOPE ..................................................................................................................... 6

2. REFERENCES ............................................................................................................. 7
   2.1 NORMATIVE REFERENCES ............................................................................... 7
   2.2 INFORMATIVE REFERENCES ........................................................................... 7

3. TERMINOLOGY AND CONVENTIONS ................................................................. 9
   3.1 CONVENTIONS ................................................................................................. 9
   3.2 DEFINITIONS .................................................................................................... 9
   3.3 ABBREVIATIONS .............................................................................................. 9

4. INTRODUCTION ........................................................................................................ 11
   4.1 VERSION 2.0.1 .................................................................................................. 11

5. DRM CONTENT FORMAT ...................................................................................... 12
   5.1 ISO BASE MEDIA FILE FORMAT ..................................................................... 12
      5.1.1 ISO File structure (INFORMATIVE) ............................................................ 12
   5.2 COMMON BOXES .............................................................................................. 13
      5.2.1 Common Headers Box ................................................................................ 13
      5.2.1.1 Common Headers Version ...................................................................... 13
      5.2.1.2 EncryptionMethod Field ........................................................................ 14
      5.2.1.3 PaddingScheme Field ............................................................................ 14
      5.2.1.4 PlaintextLength Field ............................................................................ 15
      5.2.1.5 ContentIDLength Field ......................................................................... 15
      5.2.1.6 RightsIssuerURLLength Field ............................................................... 15
      5.2.1.7 TextualHeadersLength Field .................................................................. 15
      5.2.1.8 ContentID Field ................................................................................... 15
      5.2.1.9 RightsIssuerURL Field ......................................................................... 15
      5.2.2 Textual Headers .......................................................................................... 15
      5.2.2.1 Silent header .......................................................................................... 16
      5.2.2.2 Preview header ...................................................................................... 16
      5.2.2.3 ContentURL header ............................................................................. 17
      5.2.2.4 ContentVersion header ......................................................................... 17
      5.2.2.5 Content-Locater header ....................................................................... 17
      5.2.2.6 Custom headers ..................................................................................... 18
      5.2.3 Extended Headers ....................................................................................... 18
      5.2.3.1 Group ID .............................................................................................. 18
      5.2.4 Mutable DRM Information Box .................................................................. 18
      5.2.4.1 Transaction Tracking Box ...................................................................... 19
      5.2.4.2 Rights Object Box .................................................................................. 19

6. DISCRETE MEDIA PROFILE (DCF) ........................................................................ 20
   6.1 DCF MIME TYPE ............................................................................................... 21
   6.2 DCF FILE FORMAT ............................................................................................ 21
      6.2.1 OMA Constraints on ISO Format ............................................................... 21
      6.2.2 File Branding .............................................................................................. 21
   6.3 OVERALL STRUCTURE ...................................................................................... 22
      6.3.1 OMA DRM Container Box ......................................................................... 23
      6.3.2 Discrete Media Headers Box ...................................................................... 23
         6.3.2.1 ContentType ....................................................................................... 24
         6.3.2.2 CommonHeaders ............................................................................... 24
         6.3.2.3 User-Data ........................................................................................... 24
      6.3.3 Content Object Box ................................................................................... 25
      6.3.4 Extended Boxes .......................................................................................... 26
   6.4 MULTIPLE OMA DRM CONTAINERS .............................................................. 26
      6.4.1 Referencing Multipart Objects ................................................................... 26
   6.5 ADDITIONAL EXTENSIONS ............................................................................. 27
7. CONTINUOUS MEDIA PROFILE (PDCF)........................................................................................................28

7.1 PDCF FILE FORMAT ..........................................................................................................................28

7.1.1 DRM Scheme Type ..........................................................................................................................29

7.1.2 Scheme Information ........................................................................................................................29

7.1.3 OMA DRM Key Management System .............................................................................................29

7.1.3.1 Common Headers ......................................................................................................................29

7.1.3.2 Access Unit Format Box ............................................................................................................29

7.1.4 Access Unit Format Header ............................................................................................................30

7.2 PDCF STREAMING FORMAT (INFORMATIVE) .................................................................................31

7.2.1 RTP Payload ...................................................................................................................................31

7.2.2 Session signalling ............................................................................................................................31

APPENDIX A. STATIC CONFORMANCE REQUIREMENTS (NORMATIVE) ..................................................33

A.1 CLIENT CONFORMANCE REQUIREMENTS .....................................................................................33

A.2 CLIENT CONFORMANCE REQUIREMENTS FOR THE PDCF FORMAT ............................................33

APPENDIX B. RESERVED NUMBERS (INFORMATIVE) ..............................................................................35

APPENDIX C. CHANGE HISTORY (INFORMATIVE) ..................................................................................36

C.1 APPROVED VERSION HISTORY .......................................................................................................36
Figures

Figure 1: DCF file header and body .................................................................................................................. 22
Figure 2: DCF structure ................................................................................................................................. 22
Figure 3: Data Length and IV ......................................................................................................................... 26
Figure 4: Example PDCF Structure ............................................................................................................ 28

Tables

Table 1. Algorithm-id values .......................................................................................................................... 14
Table 2. PaddingScheme values .................................................................................................................... 14
Table 3. Group ID box fields ........................................................................................................................ 18
Table 4: OMA DRM transaction tracking header field ................................................................................ 19
Table 5: OMA DRM Rights Object box fields ............................................................................................... 20
Table 6: Logical DCF box structure diagram ............................................................................................... 22
Table 7. OMA DRM Discrete Media header fields ....................................................................................... 24
Table 8: IconURI box ..................................................................................................................................... 25
Table 9: InfoURL box ..................................................................................................................................... 25
Table 10: Content Object box ....................................................................................................................... 26
Table 11 : PDCF Scheme Type for OMA DRM ............................................................................................. 29
Table 12: PDCF Scheme Version for OMA DRM ......................................................................................... 29
Table 13 : OMA DRM Headers in PDCF ....................................................................................................... 29
1. Scope

Open Mobile Alliance (OMA) specifications are the result of continuous work to define industry-wide interoperable mechanisms for developing applications and services that are deployed over wireless communication networks.

The scope of OMA “Digital Rights Management” (DRM) is to enable the distribution and consumption of digital content in a controlled manner. The content is distributed and consumed on authenticated devices per the usage rights expressed by the content owners. OMA DRM work addresses the various technical aspects of this system by providing appropriate specifications for content formats, protocols, and rights expression languages.

A number of DRM specifications have already been defined within the OMA. See [DRM-v1], [DRMCF-v1] and [DRMREL-v1] for more information.

The scope for this specification is to define the content format for DRM protected encrypted media objects and associated metadata. This specification addresses the specific format mechanisms defined in the Release 2 “Digital Rights Management” specification [DRM-v2].
2. References

2.1 Normative References


2.2 Informative References


[DRMCF-v1] “DRM Content Format”, Open Mobile Alliance™, Version 1.0, OMA-DRM-DRMCF-v1_0, URL: http://www.openmobilealliance.org/

[DRMREL-v1] “DRM Rights Expression Language”, Open Mobile Alliance™, Version 1.0, OMA-DRM-DRMREL-v1_0, URL: http://www.openmobilealliance.org/


3. Terminology and Conventions

3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119]. All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

3.2 Definitions

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Box</td>
<td>A data structure derived from the “Box” definition in [ISO14496-12].</td>
</tr>
<tr>
<td>Composite Object</td>
<td>A Media Object that contains one or more Media Objects by means of inclusion e.g. DRM messages, zip files.</td>
</tr>
<tr>
<td>Confidentiality</td>
<td>The property that information is not made available or disclosed to unauthorised individuals, entities or processes. (From [ISO 7498-2])</td>
</tr>
<tr>
<td>Content</td>
<td>One or more Media Objects.</td>
</tr>
<tr>
<td>Content Issuer</td>
<td>The entity making content available to the DRM Agent; the entity whose Content is being Protected.</td>
</tr>
<tr>
<td>Content Object</td>
<td>A single piece of Content contained in a DCF data structure. A Content Object may be DRM Content or unprotected Content.</td>
</tr>
<tr>
<td>Continuous Media</td>
<td>Content which is inherently time-based, i.e. might have an implicit or explicit duration and requires multiple iterations of an algorithm to produce a continuous media experience to a User, such as video or audio.</td>
</tr>
<tr>
<td>Device</td>
<td>A Device is a user equipment with a DRM Agent. The Device MAY include a smartcard module (e.g. a SIM) or not depending upon implementation.</td>
</tr>
<tr>
<td>Discrete Media</td>
<td>Content that can be rendered with a single pass of an algorithm to interpret the media content, media that itself does not contain an element of time, such as still images or web pages.</td>
</tr>
<tr>
<td>DRM Agent</td>
<td>The entity in the Device that manages Permissions for Media Objects on the Device.</td>
</tr>
<tr>
<td>DRM Content</td>
<td>Media Objects that are consumed according to a set of Permissions in a Rights Object.</td>
</tr>
<tr>
<td>Integrity</td>
<td>The property that data has not been altered or destroyed in an unauthorised manner.</td>
</tr>
<tr>
<td>Media Object</td>
<td>A digital work e.g. a ringing tone, a screen saver, a Java game or a Composite Object.</td>
</tr>
<tr>
<td>Permission</td>
<td>Actual usages or activities allowed (by the Rights Issuer) over DRM Content.</td>
</tr>
<tr>
<td>Rights Issuer</td>
<td>An entity that issues Rights Objects to OMA DRM Conformant Devices.</td>
</tr>
<tr>
<td>Rights Object</td>
<td>A collection of Permissions, Constraints and other attributes which define under what circumstances access is granted to, and what usages are defined for, DRM Content. All OMA DRM Conformant Devices must adhere to the Rights Object associated with DRM content.</td>
</tr>
<tr>
<td>User</td>
<td>The human user of a Device. The User does not necessarily own the Device.</td>
</tr>
</tbody>
</table>

3.3 Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3GPP</td>
<td>3rd Generation Partnership Project</td>
</tr>
<tr>
<td>4CC</td>
<td>Four Character Code</td>
</tr>
<tr>
<td>AES</td>
<td>Advanced Encryption Standard</td>
</tr>
<tr>
<td>CBC</td>
<td>Cipher Block Chaining</td>
</tr>
<tr>
<td>CEK</td>
<td>Content Encryption Key</td>
</tr>
<tr>
<td>CTR</td>
<td>Counter Mode</td>
</tr>
</tbody>
</table>
DCF  DRM Content Format
DRM  Digital Rights Management
HTTP Hypertext Transfer Protocol
ISO International Standards Organization
MIME Multipurpose Internet Mail Extensions
OMA Open Mobile Alliance
PDCF Packetized DRM Content Format
PSS Packet switched Streaming Service
RFC Request For Comments
RO Rights Object
ROAP Rights Object Acquisition Protocol
RTP Real time Transport Protocol
RTSP Real Time Streaming Protocol
SDL Syntactic Description Language
SDP Session Description Protocol
URI Uniform Resource Indicator
URL Uniform Resource Locator
4. Introduction

Within OMA DRM, Media Objects are encrypted and packaged into a specific format, the DRM Content Format (DCF). The DCF can be delivered separately from an associated Rights Object, which contains the encryption key used to encrypt the Media Object. This specification defines the DRM Content Format.

The DRM Content Format is closely related to the Rights Expression Language specification [DRMREL-v2], which defines the syntax and semantics for the Rights Objects.

In addition to encrypting the Media Object the DRM Content Format supports metadata such as

- Original content type of the media object
- Unique identifier for this DRM protected Media Object to associate it with rights
- Information about the encryption details
- Information about the rights issuing service for this DRM protected media object
- Extensions and other media type dependent metadata

The file format is extensible, so additional features may be added in the future while maintaining compatibility with the older versions. Compatibility with the version 1 Content Format [DRMCF-v1] is not maintained by this specification, therefore a different MIME type is used.

There are two profiles of the DRM Content Format. One is used for Discrete Media (such as still images) and one for Continuous Media (such as music or video). The profiles share some data structures. Both profiles are based on a widely accepted and deployed standard format, the ISO Base Media File format [ISO14496-12], but the Discrete Media profile is meant to be an all-purpose format, not aiming for full compatibility with ISO media files.

The Content Issuer can decide which profile to use for their content, but in general, the profile for Continuous Media should be used for Continuous Media content, in order to create a harmonious user experience. The Discrete Media profile should be used for other types of content. To a User, the difference is that a file conforming to the Discrete Media profile looks like a DRM protected file, whereas a file conforming to the Continuous Media profile looks and functions like a media file to the outside.

4.1 Version 2.0.1

The following table summarizes the main changes from DRM 2.0 introduced in this DRM 2.0.1 Specification.

<table>
<thead>
<tr>
<th>Title</th>
<th>Section</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content ID uniqueness</td>
<td>5.2.1.8</td>
<td>Clarifies that Content IDs must be unique per DCF and not only per Content Object.</td>
</tr>
<tr>
<td>PDCF File Format Fix and Alignment</td>
<td>7.1.3,</td>
<td>Clarifies that a PDCF may contain one OMADRMKMSBox per protected track, and in the OMADRMAUHeader moves the KeyIndicator field to enable interoperability between PDCF format and the ISMACryp specification.</td>
</tr>
<tr>
<td></td>
<td>7.1.4</td>
<td></td>
</tr>
<tr>
<td>PDCF PlainTextLength</td>
<td>5.2.1.5</td>
<td>Specifies that the PlainTextLength Common Header field must always be zero in PDCF files.</td>
</tr>
<tr>
<td>Identification of Multipart DCF and Multi-track PDCF</td>
<td>5.2.1.8</td>
<td>Specifies the method of referencing multipart DCFs and multi-track PDCFs in the ROAP Trigger &lt;contentID&gt; element.</td>
</tr>
</tbody>
</table>
5. DRM Content Format

There are two DRM Content Format profiles:

- **DCF**: The first profile is used to package and protect Discrete Media (i.e. ring tones, applications, images, etc.). The Discrete Media profile allows you to wrap any content in an envelope (DCF). That content is then encrypted as a single object agnostic of the contents internal structure and layout. This specification defines the Discrete Media format based on the types of the ISO base media file format [ISO14496-12], instead of WSP types [WSP] used in Version 1 [DRMCF-v1]. By using the ISO principles, the DCF format maintains the extensible nature of the ISO format, while keeping overhead minimal. A Device defined in [DRM-v2] MUST support the DCF format as defined in this specification. In addition, version 1 DCF as defined in [DRMCF-v1] MAY be supported.

- **PDCF**: The second profile is used to protect Continuous Media (e.g. Audio and Video). Continuous media is protected in a separate profile because it is packetized and thus the profile is called the Packetized DCF (PDCF). Applications that read and parse Continuous Media are meant to work on the file on a packet-by-packet basis. To facilitate the playback of protected Continuous Media, the storage format needs to be structured in such a way that the packets are individually protected. This structurally aware packetization is also required in order to stream Continuous Media. An OMA DRM compliant streaming server MUST be able to understand the Content Format’s structure in order to break the content into headers and packets that can be delivered to a client that understands the Content Format.

5.1 ISO Base Media File Format

The Discrete Media profile (DCF) is an object-structured file as defined in section 4 of the ISO Base Media File Format specification [ISO14496-12], but it does not include all the media-related structures due to its simplified, media agnostic design. The actual data structures and conformance to the profile is defined in this specification. If a DCF includes data structures or functionalities not conforming to this specification, a compliant file parser MAY ignore these.

The Continuous Media profile (PDCF) is fully compliant with the ISO base media file format, but this specification adds support for OMA DRM 2.0 key management on top of existing ISO derived file formats supporting encrypted media content. By default, this specification addresses the DCF format, with an additional indication if a specified data structure is also used in the PDCF format.

5.1.1 ISO File structure  (INFORMATIVE)

This section is informative and is based on the ISO Base Media File Format specification [ISO14496-12].

The ISO base media file format is structured around an object-oriented design of boxes. A basic box has two mandatory fields, *size* and *type*. The type identifier is used to dynamically bind a box to a statically defined type and the *size* is an offset from start to the end of the box. A Box type identifier is a *Unique Identifier Number*. List of reserved numbers can be found in Appendix B. The identifier is constructed from four bytes, each representing a human-readable character, thus the name *Four Character Code* (4CC).

The ISO base format uses a language called Syntax Description Language (SDL) for defining data structures. SDL has similarities with some programming languages and supports object orientation. The box class is the superclass for all structures containing data in the file format.

A basic box is defined as:

```
aligned(8) class Box (unsigned int(32) boxtype, optional unsigned int(8)[16] extended_type) {
    unsigned int(32) size;
    unsigned int(32) type = boxtype;
    if (size==1) {
        unsigned int(64) largesize;
    } else if (size==0) {
        // box extends to end of file
    }
    if (boxtype=='uuid') {
```
Box alignment is by default to the next byte boundary in the end of the box. Extra padding should not be needed as all data types in e.g. the DCF are terminated on byte boundaries.

Since one of the design goals for the DCF is extensibility, it is important to carry version information with each data type. The ISO specification has a predefined type to support this, the FullBox, which is derived from the simple Box base class.

Extending a parent class has similar semantics as in many programming languages; the parent class data members precede the child class definitions. A representation of the FullBox above is:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>unsigned int(32)</td>
<td>Offset to the end of the box</td>
</tr>
<tr>
<td>Type</td>
<td>unsigned int(32)</td>
<td>Box type 4CC</td>
</tr>
<tr>
<td>Version</td>
<td>unsigned int(8)</td>
<td>Version field</td>
</tr>
<tr>
<td>Flags</td>
<td>unsigned int(24)</td>
<td>Additional flags</td>
</tr>
</tbody>
</table>

The numeric fields in the ISO format are in network byte order.

5.2 Common Boxes

5.2.1 Common Headers Box

The Common Headers box defines a structure for the required headers. Their semantics are defined in the sections below. This box MUST appear in both DCF and PDCF. This box includes the mandatory headers as fixed fields and provides a mechanism to insert additional headers as arbitrary name value pairs. For application in DCF and PDCF, see sections 6.3.2 and 7.1.3.1 for details.

A Device MUST NOT modify any of the fields in the Common Headers box.

5.2.1.1 Common Headers Version

The version field of the FullBox defines which version of DRM Content Format specification was used by the author of the Content Object. The value for version MUST be 0 for objects conforming to this specification.
5.2.1.2 EncryptionMethod Field

The EncryptionMethod field defines how the encrypted content can be decrypted. Values for the field are defined in the table below.

Table 1. Algorithm-id values

<table>
<thead>
<tr>
<th>Algorithm-id</th>
<th>Value</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>NULL</td>
<td>0x00</td>
<td>No encryption for this object. NULL encrypted Content Objects may be used without acquiring a Rights Object. Value of the PaddingScheme field MUST be 0.</td>
</tr>
<tr>
<td>AES_128_CBC</td>
<td>0x01</td>
<td>AES symmetric encryption as defined by NIST [AES]. 128 bit keys. Cipher block chaining mode (CBC). 128 bit initialization vector prefixing the ciphertext (for non-streamable PDCF files this is included in the OMADRMAUHeader). Padding according to RFC 2630.</td>
</tr>
<tr>
<td>AES_128_CTR</td>
<td>0x02</td>
<td>AES symmetric encryption as defined by NIST [AES]. 128 bit keys. Counter mode (CTR). 128 bit initial counter value prefixes the ciphertext (for non-streamable PDCF files this is included in the OMADRMAUHeader). For each cipherblock the counter is incremented by 1 (modulo 2^{128}). No padding.</td>
</tr>
</tbody>
</table>

Rights Issuers should take care in using NULL EncryptionMethod because, given a null-encrypted Media Object within a DCF or PDCF, the following statements hold true:

- Null-encrypted Media Objects do not have any Confidentiality protection.
- Null-encrypted Media Objects can always be used without an associated Rights Object.
- Null-encrypted Media Objects may not have any integrity protection.

5.2.1.3 PaddingScheme Field

The PaddingScheme parameter defines how the last block of ciphertext is padded. Values of the PaddingScheme field are defined in the table below:

Table 2. PaddingScheme values

<table>
<thead>
<tr>
<th>Padding-Scheme</th>
<th>Value</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>0x00</td>
<td>No padding (e.g. when using NULL or CTR algorithm).</td>
</tr>
<tr>
<td>RFC_2630</td>
<td>0x01</td>
<td>Padding according to RFC 2630.</td>
</tr>
</tbody>
</table>
5.2.1.4 PlaintextLength Field

The *PlaintextLength* field defines the length of the original plaintext. In the case of DCF and if the content is encrypted, it MUST have a PlaintextLength value set. If the extracted content length does not match the PlaintextLength field value, it is an error and the Content Object MUST be discarded. In a progressive download scenario, the DRM Agent can verify the PlaintextLength only after the complete Content Object has been received and possibly after content use has started.

In the case of PDCF the PlaintextLength MUST be zero. DRM Agents should ignore the PlaintextLength for PDCF files; instead they should use the relevant ISO Base Media boxes to identify track properties.

5.2.1.5 ContentIDLength Field

The *ContentIDLength* field defines the number of bytes occupied by the *ContentID* field. The value MUST be greater than zero. A Device MUST support ContentIDs of at least 256 bytes. For best interoperability, content author should not use a ContentID larger than 256 bytes.

5.2.1.6 RightsIssuerURLLength Field

The *RightsIssuerURLLength* field indicates the number of bytes occupied by the RightsIssuerURL field. A Device MUST support RightsIssuerURLs of at least 256 bytes. For best interoperability, content author should not use a RightsIssuerURL larger than 256 bytes.

5.2.1.7 TextualHeadersLength Field

The *TextualHeadersLength* field indicates the number of bytes occupied by the *TextualHeaders* field. Although it is possible with this version of the parent box to implicitly determine the TextualHeaders field length from the box length, this might not be the case in future versions. Thus, conforming tools MUST use the TextualHeadersLength field. A Device MUST support textual headers of at least 2048 bytes total length.

5.2.1.8 ContentID Field

The *ContentID* field MUST contain a globally unique identifier for this Content Object. Note that even if two or more DCFs contain the same Content, the Content Objects will each have a different (and globally unique) ContentID. The value MUST be encoded using US-ASCII encoding.

The value MUST be a unique URI according to [RFC2396]. The use of globally unique ContentID’s is required for OMA DRM and it is the responsibility of the content author to guarantee the uniqueness of the ContentID within their own namespace.

If the Content Object is referenced from a DRM Rights Object, the value of the *ContentID* field MUST match the value of the referencing element of the Rights Object as defined in [DRMREL-v2]. The ContentID MUST be in the ‘cid-url’ format of [RFC2392].

5.2.1.9 RightsIssuerURL Field

The *RightsIssuerURL* field defines the Rights Issuer URL. The Rights Issuer URL MAY be used by the consuming Device to obtain Rights for this DRM Content. The mechanism is defined in OMA DRM specification [DRM-v2]. The value of the RightsIssuerURL field MUST be encoded using US-ASCII encoding. The length of this field is indicated by the RightsIssuerURLLength field.

The value of the RightsIssuerURL MUST be a URL according to [RFC2396], and MUST be an absolute identifier. The RightsIssuerURL MAY be empty e.g. if the Content Object is not encrypted.

5.2.2 Textual Headers

The *TextualHeaders* field MAY contain additional information about the content.

Textual headers are represented by name value pairs, where name and value are separated with a colon ‘:’ and the pair is terminated with a NULL (‘\0’) character. A header (name value pair) MUST NOT include leading or trailing whitespace.
(such as \n). Further, a header name MUST NOT include a colon (\:) character, as the first instance or the character will stop scanning for the header name. Header value MAY include colon characters as the value is always assumed to continue after the first colon until a NULL character is reached.

The next header name MUST begin immediately after the terminating NULL character of the previous header, if TextualHeadersLength is greater than the current scanning position. All headers MUST have a value, i.e. an empty value is not permitted.

The textual headers field continues until the TextualHeadersLength offset or the end of the box is reached. The TextualHeadersLength field MUST be used to determine the TextualHeaders field length.

An example representation of the textual headers:

*Silent:on-demand;http://myissuer.com/silent?cid=428*0
*Preview:instant;cid:429@myissuer.com*0

Each supported header is defined using augmented Backus-Naur Form (BNF) [RFC2234]. The textual headers are encoded using UTF-8 encoding. Ordering of headers is significant, and the headers MUST be in the order of priority, from highest to lowest. This means that e.g. if the textual headers include both Silent and Preview headers, whichever appears first in the field is considered to have priority over the second.

### 5.2.2.1 Silent header

The Silent header is an indication to the client that the Rights Object for this DRM Content can be obtained silently from the Rights Issuer, without user interaction for payments, etc.

```
Silent = "Silent" "\:" silent-method "\:" parameter
silent-method = token
parameter = silent-rights-url
silent-rights-url = token
```

<table>
<thead>
<tr>
<th>silent-method</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>“on-demand”</td>
<td>Rights should be acquired silently, on demand when the user chooses to play the content.</td>
</tr>
<tr>
<td>“in-advance”</td>
<td>Rights should be acquired in advance, at the earliest opportunity.</td>
</tr>
</tbody>
</table>

The parameter silent-rights-url MUST be a URL according to [RFC2396] and a successful request to the URL MUST return a ROAP Trigger, a Download Descriptor or a bundled Download Descriptor and ROAP Trigger as defined in [DRM-v2]. If silent-rights-url is a HTTP URL and the request fails with error code 404 Not Found [RFC2616], the Device SHOULD NOT make further requests to the URL. If the request fails with some other error, the Device MAY retry the request at a later time.

The parameter silent-rights-url MUST be specified on the Silent header. The device MUST use this silent-rights-url to obtain rights silently and automatically according to [DRM-v2].

### 5.2.2.2 Preview header

The Preview header contains an indication to the client that it is possible to provide a preview for this DRM Content.

If the preview-method is “instant”, then the specific media element to be used for preview MUST be indicated using the preview-element-uri parameter. In addition, this media element MUST be NULL-encrypted, and as such, MUST have an EncryptionMethod header with the algorithm-id parameter set to NULL.

```
Preview = "Preview" "\:" preview-method "\:" parameter
preview-method = token
parameter = preview-element-uri | preview-rights-url
preview-element-uri = token
preview-rights-url = token
```

© 2008 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.
<table>
<thead>
<tr>
<th>Preview-method</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;instant&quot;</td>
<td>This indicates that one of the elements within this file can be used for preview. If instant method is specified, then preview-element-uri MUST be specified.</td>
</tr>
<tr>
<td>&quot;preview-rights&quot;</td>
<td>This indicates that a preview Rights Object can be obtained by requesting it silently from the Rights Issuer, without user interaction. If preview-rights method is specified, then preview-rights-url MUST be specified.</td>
</tr>
</tbody>
</table>

The parameter `preview-element-uri` MUST be a unique identifier and a URI according to [RFC2396]. And, it MUST resolve to an element present within the same file.

The parameter `preview-rights-url` MUST be a URL according to [RFC2396] and a successful request to the URL MUST return a ROAP Trigger, a Download Descriptor or a bundled Download Descriptor and ROAP Trigger as defined in [DRM-v2]. If `preview-rights-url` is a HTTP URL and the request fails with error code 404 Not Found [RFC2616], the Device SHOULD NOT make further requests to the URL. If the request fails with some other error, the Device MAY retry the request at a later time.

If the `preview-method` is indicated as “instant”, the preview element can be used freely with unlimited use, without acquiring any Rights Objects.

If the `preview-method` is “preview-rights”, then the `preview-rights-url` MUST be indicated as a parameter. When the client connects to the Rights Issuer with this URL, this MUST NOT result in any re-direction.

### 5.2.2.3 ContentURL header

The `ContentURL` header is used to indicate a location for acquiring the DCF or PDCF. This MAY be used to e.g. download an alternative version of the file if a device does not support the content types in the current file, such as resolution or codec. The consuming device MAY provide the option to forward the ContentURL to other users as an alternative form of superdistribution. The mechanism is defined in OMA DRM specification [DRM-v2].

```plaintext
ContentURL = "ContentURL" : content-url
content-url = token
```

The `content-url` MUST be a URL according to [RFC2396] and MUST be an absolute identifier. The device MAY access the ContentURL from the DCF and use it to establish e.g. a browsing session without acquiring a Rights Object for the DRM Content.

### 5.2.2.4 ContentVersion header

The `ContentVersion` header defines the version of the content. This header MAY be used to uniquely identify the incarnation of this DRM Content Object.

```plaintext
ContentVersion = "ContentVersion" : original-content-identifier : version-identifier
original-content-identifier = token
version-identifier = *digit
```

Where `original-content-identifier` MUST be a matching string for all versions of the same Content and `version-identifier` MUST be a number in range 0..65535, incremented by each version.

### 5.2.2.5 Content-Location header

The `Content-Location` header MAY be used to indicate a relative location for the Content Object. This MAY be used for e.g. referencing purposes within the DCF file or determining a meaningful file name when exporting the Content Object.

```plaintext
ContentLocation = "Content-Location" : content-uri
content-uri = token
```

The `content-uri` MUST be a file name, relative to the location of the DCF file.
5.2.2.6 Custom headers

Content author MAY insert additional Custom headers to the TextualHeaders field. Custom headers MUST follow the generic syntax defined below, encoded using UTF-8 encoding.

| OtherHeader = Header-name ":" Header-value |
| Header-name = token                        |
| Header-value = token                       |

Consuming Devices MUST ignore the headers that they do not recognize.

5.2.3 Extended Headers

The ExtendedHeaders field MAY include zero or more nested boxes that add functionalities to the common headers. The ExtendedHeaders field continues until the end of the parent box is reached.

5.2.3.1 Group ID

The ExtendedHeaders field MAY include one instance of the OMA DRM GroupID Box:

```c
aligned (8) class OMADRMGroupID extends FullBox('grpi', version, 0) {
    unsigned int(16) GroupIDLength;   // length of the Group ID URI
    unsigned int(8) GKEncryptionMethod;  // Group Key encryption algorithm
    unsigned int(16) GKLength;   // length of the encrypted Group Key
    char GroupID[GroupIDLength];    // Group ID URI
    byte GroupKey[GKLength];       // Encrypted Group Key and encryption information
}
```

The GroupID value identifies this DCF as part of a group of DCFs whose Rights can be defined in a common group Rights Object instead of (or in addition to) in separate content-specific Rights Objects. The value of GroupID MUST be a URI according to [RFC2396] and MUST contain a globally unique identifier. The value MUST be encoded using US-ASCII encoding.

Generally each content item in a group will be encrypted with a different content item encryption key. A single additional key (used for the whole group) is used to encrypt each content item encryption key for storage in the GroupKey field. This single key is the value of the CEK in an associated group RO. Note that since the Group ID box is part of the OMA DRM container box, it is possible for different content items in a multipart DCF to belong to different groups. The GKEncryptionMethod field defines the algorithm used to encrypt the content item encryption keys, as defined in Section 5.2.1.2 and it defines the structure of the GroupKey field that can contain, next to the actual encrypted content item encryption key (referred to in Section 5.2.1.2 as ‘ciphertext’), additional information such as initialization vector or initial counter value. The NULL EncryptionMethod MUST NOT be used as a GKEncryptionMethod.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>GroupIDLength</td>
<td>unsigned int(16)</td>
<td>Length of the Group ID URI field</td>
</tr>
<tr>
<td>GKEncryptionMethod</td>
<td>unsigned int(8)</td>
<td>Group Key encryption algorithm</td>
</tr>
<tr>
<td>GKLength</td>
<td>unsigned int(16)</td>
<td>Length of the GroupKey field</td>
</tr>
<tr>
<td>GroupID</td>
<td>char[]</td>
<td>Group ID URI</td>
</tr>
<tr>
<td>GroupKey</td>
<td>byte[EncryptedGKLength]</td>
<td>Encrypted Group Key and additional encryption information such as initialization vector, counter values, padding as defined in Section 5.2.1.2</td>
</tr>
</tbody>
</table>

5.2.4 Mutable DRM Information Box

The Mutable DRM Information box MAY appear in both DCF and non-streamable PDCF. Mutable DRM Information is not applicable to streamable PDCF. In the OMA DRM system, the
MutableDRMInformation box is used to include information editable by the Device, and thus is not protected for integrity. A Device MUST ignore the MutableDRMInformation box when calculating the DCF hash.

The MutableDRMInformation box MUST be located at the top level of the box hierarchy and there MUST NOT be more than one instance of the box per DCF or PDCF. The MutableDRMInformation box MAY include free space boxes as defined in ISO base media file format [ISO14496-12] to pre-allocate space for editing. A MutableDRMInformation box MUST NOT appear in the beginning of the file, but MAY appear after the last OMADRMContainer (see 6.3.1) in DCF and after the movie box in PDCF. Having the MutableDRMInformation box as the last box in the file is RECOMMENDED for DCF.

For PDCF the location of the MutableDRMInformation should be carefully considered by the Content Issuer. Generally it is preferable to place the MutableDRMInformation directly after the Movie Box; this enables Transaction Tracking and Rights Object delivery during Progressive Download. However, if the MutableDRMInformation box is after the movie box it will be difficult for the client to insert new Rights Objects into the Rights Object box because if the size of the MutableDRMInformation box changes the device must also update the Chunk Offset Box (‘stco’) in the PDCF headers. Therefore Content Issuers are recommended to also include a Free Space Box if the MutableDRMInformation is placed before the Media Data.

A Device MAY modify, extend, truncate, delete or add the MutableDRMInformation box. The contents of the box MUST be interpreted as an array of Boxes, continuing until the end of the parent box.

```
aligned(8) MutableDRMInformation extends Box('mdri') {
    Box data[];  // array of any boxes and free space
}
```

### 5.2.4.1 Transaction Tracking Box

The OMA DRM Transaction Tracking Box enables transaction tracking as defined in [DRM-v2] section 12.3. The OMADRMTransactionTracking box MUST include a single TransactionID value as defined below. It MAY appear in both DCF and PDCF.

```
aligned(8) class OMADRMTransactionTracking extends FullBox('odtt', 0, 0) {
    char TransactionID[16];  // value to enable transaction tracking
}
```

<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>TransactionID</td>
<td>char[16]</td>
<td>TransactionID of the DCF or PDCF respectively</td>
</tr>
</tbody>
</table>

The Rights Issuer MAY provide any value as a TransactionID to the DRM Agent during the Rights acquisition process and the TransactionID included in the DRM Container may be changed by the DRM Agent as defined in [DRM-v2]. When packaging content, the TransactionID MAY be set to an arbitrary value.

As per [DRM-v2] section 12.3 the DRM Agent does not need to generate the OMADRMTransactionTracking box, nor does the DRM Agent ever need to modify the size of the box.

### 5.2.4.2 Rights Object Box

The rights object box MAY be used to insert a Protected Rights Object, defined in [DRM-v2] section 5.3.8, into a DCF or PDCF. A MutableDRMInformation box MAY include zero or more Rights Object boxes. The Rights Object is treated as binary data and a Device MAY add or delete Rights Object boxes in the MutableDRMInformation box.

```
aligned(8) class OMADRMRightsObject extends FullBox('odrb', 0, 0) {
    byte Data[];  // binary Rights Object
}
```
Table 5: OMA DRM Rights Object box fields

<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>byte[]</td>
<td>A Rights Object as binary data</td>
</tr>
</tbody>
</table>

5.3 DCF Hash Calculation

Content Objects MAY be protected for integrity by including a DCF hash into a Rights Object or ROAP request. Since (P)DCF MAY include structures editable by the Device, these structures are excluded from hash calculation. The DCF hash MUST be calculated from the beginning of the DCF to the end of the last OMADRMContainer, ignoring the MutableDRMInformation box. PDCF hash MUST be calculated from the beginning of the PDCF, skipping the MutableDRMInformation box after the movie box, or end of file in case there is no MutableDRMInformation box present.
6. Discrete Media Profile (DCF)

This section defines the DRM Content Format for Discrete Media.

6.1 DCF MIME Type

The MIME type for objects conforming to the format defined in this section MUST be

```
application/vnd.oma.drm.dcf
```

and the corresponding file extension MUST be ".odf".

6.2 DCF File Format

The structure of the Discrete Media profile of DRM Content Format (DCF) MUST be according to the structure definitions below.

A DCF file MUST include at least one OMA DRM Container box. The OMA DRM Container box is a container for a single Content Object and its associated headers. It MUST appear on the top level, i.e. to conform to this specification, it MUST NOT be nested inside another data type. There MAY exist multiple OMA DRM Container boxes in a file, but one MUST immediately follow the file header, and they all MUST be located on the top level in the nesting structure.

The version indicator field in each box MUST be 0 for files conforming to this specification. All numeric fields in the format MUST be stored in network byte order.

6.2.1 OMA Constraints on ISO Format

In files conforming to this specification, box size MUST be greater than 1 unless otherwise specified and the extended_type MUST NOT be used in the mandatory boxes. Some of the mandatory boxes MUST support the 64 bit length field and for those boxes, size field MUST be set to 1. Also note that in some earlier ISO specifications, the term atom was used to describe the file format structures, but the structures specified in this specification are called boxes in order to be consistent with current specifications.

The FullBox version is typically started from zero (0), incremented by each revision. The flags field MAY be used to include additional information, but SHOULD normally be set to 0, unless otherwise specified. This specification names each supported box to indicate that a box has a defined structure and a purpose in the OMA DRM Content Format.

There are also placeholders for extensions, with only a generic box reference. These extensions may be defined later, and thus a conforming file parser SHOULD skip any extension boxes it does not understand. In addition, all of the toplevel boxes are derived from the FullBox type, which supports version information. Later specifications MAY increment the version number if changes are made to any common data structures. Later versions of the boxes defined in this specification should remain backwards compatible with the help of this version indicator. A parser conforming to this specification MAY attempt to parse a box which has a greater version number than this specification, but the conformance is limited to the current version (0) of this specification. A conforming parser MUST check the version number field.

6.2.2 File Branding

The ISO base media file format defines a File Type box for identifying the major brand of the media file along with compatible brands. Files conforming to the Discrete Media profile MUST include a File Type box with the DCF brand as the major brand number and compatible brand to make the File Type box fixed length. The DCF major brand is 32 bits (4 octets) wide with the hexadecimal value 0x6f646366 ("odcf"). This MUST be followed by a four-octet minor version indicator and the DCF brand as the single compatible brand, making the file header a total of 20 octets (160 bits) from the beginning of the file. The minor version field is in network byte order. For files conforming to this version of the DCF specification the version value MUST be 2 (0x00000002). A conforming file parser MUST support the minor version number. It should be noted that future minor versions of the DCF file format might use more compatible brands in the File Type box, changing the file header length. The Figure 1 shows the relationship of the File Type, brand, version and rest of the file content.
6.3 Overall structure

The high-level overview of the DCF format is depicted in the Figure 2. The mandatory parts of the format include the file header (File Type box with brand number and minor version fields), immediately followed by an OMA DRM Container box. The OMA DRM Container box MUST include a DCF headers box and a DRM Content box.

The design principles for the format include that the DCF headers box is located at a fixed offset from the beginning of the file, and thus, the OMA DRM Container box MUST be the first box after the file header of 20 octets and the DCF headers box MUST be the first box in the OMA DRM Container.

The table below outlines the mandatory boxes and their order. Additional boxes MAY be added after the mandatory boxes have first appeared. Table 6 shows the nesting order of the mandatory boxes, on the left is the parent and on the right, the child. The first column indicates which fields and boxes MUST be present in DCF (marked as ‘M’) and which boxes MAY appear in the DCF (marked as ‘O’). Note that in the table, the second OMA DRM Container box MUST include all the mandatory nested boxes as well.

<table>
<thead>
<tr>
<th>Present in DCF</th>
<th>Data type/value</th>
<th>Nesting level</th>
<th>Offset from beginning of file</th>
<th>Field purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>Box(‘ftyp’)</td>
<td>0</td>
<td>0</td>
<td>File header (fixed File Type box, 20 bytes)</td>
</tr>
<tr>
<td>M</td>
<td>Box(‘odrm’)</td>
<td>0</td>
<td>20</td>
<td>OMA DRM Container box</td>
</tr>
</tbody>
</table>

© 2008 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.
6.3.1 OMA DRM Container Box

```java
aligned(8) class OMADRMContainer extends FullBox('drm', version, 0) {
    OMADRMDiscreteHeaders ContentHeaders; // Headers for Discrete Media DCF
    OMADRMContentObject DRMContent; // Actual encrypted content
    Box Extensions[]; // Extensions, to the end of the box
}
```

The OMA DRM Container box MUST include a single OMADRMDiscreteHeaders box and a single OMADRMContentObject box, followed by optional extensions. The Extensions inside the OMA DRM Container box are defined by OMA. The OMA DRM Container box MUST support 64 bit length attributes, i.e. the size attribute MUST be set to 1, and largestsize MUST be used for determining the box size.

6.3.2 Discrete Media Headers Box

```java
aligned(8) class OMADRMDiscreteHeaders extends FullBox('odhe', version, flags) {
    unsigned int(8) ContentTypeLength; // Content Type Length
    char ContentType[]; // Content Type String
    OMADRMCommonHeaders CommonHeaders; // Common headers (same as with PDCF)
    if(flags & 0x000001) {
        UserDataBox UserData; // ISO User Data Box (optional)
    }
}
```

The Discrete Media headers box includes fields specific to the DCF format and the Common Headers box, followed by an optional user-data box. There MUST be exactly one OMADRMDiscreteHeaders box in a single OMA DRM Container box, as the first box in the container.

The ContentType field indicates the actual media type contained in the OMA DRM container. There MUST be exactly one OMADRMCommonHeaders (see section 5.2.1 for details) box per a single OMADRMDiscreteHeaders box.
Table 7. OMA DRM Discrete Media header fields

<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>ContentTypeLength</td>
<td>Unsigned int(8)</td>
<td>Length of the ContentType field</td>
</tr>
<tr>
<td>ContentType</td>
<td>ContentTypeLength octets</td>
<td>The MIME media type of the plaintext data encoded as US-ASCII</td>
</tr>
<tr>
<td>CommonHeaders</td>
<td>OMADRMCommonHeaders</td>
<td>OMA DRM Common Headers box as in 5.2.1</td>
</tr>
<tr>
<td>UserData</td>
<td>UserDataBox</td>
<td>User Data as defined in 6.3.2.3 (OPTIONAL)</td>
</tr>
</tbody>
</table>

6.3.2.1 ContentType

The ContentType field MUST indicate the original MIME media type of the Content Object i.e. what content type the result of a successful extraction of the OMADRMContent box represents. The ContentType field is encoded using US-ASCII encoding and MUST NOT include a NULL character.

6.3.2.2 CommonHeaders

The CommonHeaders field MUST be the same box as defined in 5.2.1.

6.3.2.3 User-Data

A user-data box (‘udta’), as defined in [ISO14496-12], MAY be present in the discrete headers box. When a DCF includes the UserDataBox, it MUST be added immediately after the OMADRMCommonHeaders box. The presence of the user-data box MUST be indicated with the flag 0x000001 in the containing box header. The user-data box is a container box for informative user data. This user information is formatted as a set of sub-boxes with specific box types that more precisely define their usage. Each of the sub-boxes MAY be included only once unless otherwise noted.

Some of these sub-boxes contain text information, which is metadata, as defined in [TS26.244]. This specification supports a subset of the sub-boxes defined in [TS26.244].

6.3.2.3.1 Title

The Title box (‘titl’) contains a descriptive name for this Content Object, as defined in [TS26.244]. The title is only informative and the device MAY use it e.g. to derive a filename when the DRM protected object is received and stored into a local repository. Other names may be transmitted outside this object (e.g. Content-Disposition header in HTTP) and they may override the name specified in this element.

This box MAY be included zero or more times using different language codes. The syntax for this box is the same as defined in [TS26.244]. A Device MUST support UTF-8 encoded text and MAY support UTF-16 encoded text.

6.3.2.3.2 Description

The Description box (‘dscp’) contains a description of the Content Object, as defined in [TS26.244]. This text is informative and the device MAY display it to the user prior to acquiring Rights for the Content Object.

This box MAY be included zero or more times using different language codes. The syntax for this box is the same as defined in [TS26.244]. A Device MUST support UTF-8 encoded text and MAY support UTF-16 encoded text.

6.3.2.3.3 Copyright

The Copyright box (‘cprt’) contains a copyright declaration of the organization holding the copyright of the Content Object, as defined in [TS26.244]. This text is informative and the device MAY display it to the user prior to acquiring Rights for the Content Object.

This box MAY be included zero or more times using different language codes. The syntax for this box is the same as defined in [TS26.244]. A Device MUST support UTF-8 encoded text and MAY support UTF-16 encoded text.
6.3.2.3.4 **Author**

The Author box (‘auth’) contains a textual string representing the author of the Content Object as defined in [TS26.244]. This text is informative and the device MAY display it to the user prior to acquiring Rights for the Content Object.

This box MAY be included zero or more times using different language codes. The syntax for this box is the same as defined in [TS26.244]. A Device MUST support UTF-8 encoded text and MAY support UTF-16 encoded text.

6.3.2.3.5 **IconURI**

```c
aligned(8) class OMADRMIconURI extends FullBox('icnu', version, 0) {
    char IconURI[];  // Icon URI
}
```

The IconURI box (‘icnu’) contains a URI where an appropriate icon for this content may be retrieved from. The device MAY request the object at this URI, and if an appropriate content is returned, use this as an icon associated with the content to the user.

The value of the **IconURI** MUST be a URI according to [RFC2396]. It is a string encoded using UTF-8 characters, continuing until the end of the box is reached.

If the DCF is a Multipart DCF, a **IconURI** MAY be a CID reference [RFC2557] within the current file. In this case, the referenced Content Object MUST be NULL-encrypted.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>IconURI</td>
<td>char[]</td>
<td>URI for an Icon for the content.</td>
</tr>
</tbody>
</table>

6.3.2.3.6 **InfoURL**

```c
aligned(8) class OMADRMInfoURL extends FullBox('infu', version, 0) {
    char InfoURL[];  // Info URL
}
```

The InfoURL box (‘infu’) contains a URL where additional information can be found regarding the Content Object. The device MAY obtain this information prior to using the **RightsIssuerURL** field or after the Rights Object has been obtained.

The value of the **InfoURL** MUST be a URL according to [RFC2396] and MUST be an absolute identifier. It is a string encoded using UTF-8 characters, continuing until the end of the box is reached.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>InfoURL</td>
<td>char[]</td>
<td>Location of additional information for the content.</td>
</tr>
</tbody>
</table>

6.3.3 **Content Object Box**

```c
aligned(8) class OMADRMContentObject extends FullBox('odda', version, 0) {
    unsigned int(64) OMADRMDataLength;  // Length of the encrypted content
    byte OMADRMData[];                 // Encrypted content
}
```

The Content Object box MUST include only the data length field and data bytes for a single Content Object. Later revisions of this box may include additional fields, so conforming implementations MUST use the **OMADRMDataLength** field to indicate/determine the amount of actual data bytes. The data length includes the Initialization Vector in the beginning of the encrypted data, as depicted in Figure 3.
The Content Object box MUST support the 64 bit size field and thus size MUST be set to 1 and largesize MUST be used for determining actual box size. The OMA DRMDataLength field MAY indicate a length of zero, and the Device MAY try to acquire the actual Content Object by using e.g. the ContentURL, if provided.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>OMA DRMDataLength</td>
<td>Unsigned int(64)</td>
<td>Length of the OMA DRMData field, in octets</td>
</tr>
<tr>
<td>OMA DRMData</td>
<td>byte []</td>
<td>Content bytes, as specified by the OMA DRMDiscreteHeaders box</td>
</tr>
</tbody>
</table>

### 6.3.4 Extended Boxes

Any additional boxes contained in a single OMA DRM container box have not been defined in this specification. A Content Issuer MAY place additional boxes into the Extensions but Devices MAY ignore these.

### 6.4 Multiple OMA DRM Containers

A DCF MAY include more than one OMA DRM Container. Each of these containers MUST conform to the definition of the OMA DRM Container, and MUST be placed sequentially on the top level (i.e. nesting them is not allowed). The media type of Content Object in each these containers MAY BE different. However, the media type of the first OMA DRM Container is considered to be the default media type of the DCF’s content.

Each OMA DRM Container MUST have a unique ContentID in its headers. This kind of a DCF with multiple Content containers is called a Multipart DCF.

Note that a Multipart DCF is different from a DCF including a Composite Object. A Composite Object (such as MIME multipart, ZIP and so on) is included in a single OMA DRM Container and has only one set of OMA DRM headers associated with it, whereas Multipart DCFs contain multiple OMA DRM Containers each including separate headers associated with the contained content. Multipart DCFs support the association of different rights with individual Media Objects.

### 6.4.1 Referencing Multipart Objects

As each object in the Multipart DCF has its own ContentID and MAY have a Content-Location header, the CID mechanism from [RFC2557] or the Content-Location mechanism from [RFC2616] MUST be used for referencing objects within the Multipart DCF. The reference MAY then be used in e.g. multimedia presentations to include objects from within the Multipart DCF. Individual Content Objects cannot be referenced from e.g. presentations outside the DCF file.
The ContentID is considered to be internal for the DRM Content Format and DRM Agent, and ContentIDs are referenced from outside the DRM Content Format only to associate it with a Rights Object. Transport protocols MUST define their own mechanisms how to reference to a DRM Content Format file.

6.5 Additional Extensions

Additional extension boxes MAY be added after the first OMA DRM Container. A conforming file parser, which does not recognize the additional boxes, MUST ignore them. However, any extensions MUST be designed in a way that the mandatory parts of this specification are always included and the file remains interoperable with conforming implementations.
7. Continuous Media Profile (PDCF)

The Continuous (Packetized) Media profile is targeted for media content like audio and video. Audio and video files MAY be included in a DCF format, but since the PDCF format has been specifically designed for Continuous Media, it provides additional advantages for those media types.

The PDCF format is an instance of the ISO Base Media File Format [ISO14496-12] that supports encrypted media tracks, which MUST use OMA DRM for key management and MUST include the OMA DRM data structures defined in this specification. Examples of ISO Base Media File Format instantiations are the 3GP format [TS26.244] and 3G2 format [C.S0050].

The PDCF format MAY be used for downloaded content or for hosting streamable content. OMA DRM specifies common data structures for file formats and additional information on top of streaming services. The OMA DRM 2.0 specifications define key management functionality supporting Continuous Media but services can optimise the protocols and codecs in their architecture. Supporting the PDCF format is OPTIONAL for a Device.

7.1 PDCF File format

This specification defines the OMA DRM key management part of the PDCF format. In the ProtectionSchemeInfoBox, there is space for a “black box” (SchemeInformationBox) describing the key management governing access to the encrypted media content. In a PDCF file, this box MUST be the OMA DRM KMS Box.

The basic PDCF file format data structures are defined by the corresponding base file format specification, and this specification only adds OMA DRM specific structures and parameters. Other DRM mechanisms MAY be used in those file formats supporting encrypted media tracks, but not in PDCF files, as explained in this specification.

Figure 4: Example PDCF Structure

The Figure 4 illustrates how protection information is stored in a PDCF. It is an example where only the video track is protected by placing a ProtectionSchemeInfoBox into the track and specifying the OMA DRM identifier as the key management system. All tracks in a PDCF can be protected with the mechanism.

There is a difference between a streamable PDCF and a non-streamable PDCF. A streamable PDCF MUST conform to the server profile of the file format specification, and the media data is stored as packets. In a non-streamable PDCF, media data is stored as samples. An access unit is a group of one or more samples.

The encryption process changes both packet and sample formats from the original plaintext. The file format may support also other DRM key management systems than OMA DRM 2.0, but the encrypted access unit format is specific to OMA DRM. Thus, in the encryption process, non-streamable PDCFs MUST have the OMA DRMAUHeader (see 7.1.4) inserted before each access unit.
7.1.1 DRM Scheme Type

The SchemeTypeBox includes information on which DRM system is being used to manage keys and decryption of the content. As the media file format MAY support also other key management systems than OMA DRM, the key management system in use is indicated by a 4CC in the SchemeType field [ISO14496-12].

<table>
<thead>
<tr>
<th>SchemeType</th>
<th>Value</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>OMA DRM</td>
<td>'odkm'</td>
<td>OMA DRM is used for key management in the PDCF.</td>
</tr>
</tbody>
</table>

Table 11: PDCF Scheme Type for OMA DRM

<table>
<thead>
<tr>
<th>SchemeVersion</th>
<th>Value</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>OMA DRM 2.0</td>
<td>0x00000200</td>
<td>OMA DRM version is 2.0</td>
</tr>
</tbody>
</table>

Table 12: PDCF Scheme Version for OMA DRM

For PDCF files conforming to this specification, the SchemeType MUST be the 4CC 'odkm', and SchemeVersion MUST be 0x00000200 (version 2.0). If OMA DRM key management scheme 'odkm' is indicated, then the file is a PDCF and MUST contain at least one OMADRMKMSBox. A PDCF MUST support only OMA DRM for the key management system.

7.1.2 Scheme Information

The SchemeInformationBox (‘schi’) is used to carry DRM key management system specific information, thus it is only a container box. For OMA DRM, this box MUST include exactly one OMADRMKMSBox, as the first sub-box.

7.1.3 OMA DRM Key Management System

There MAY be several instances of the OMADRMKMSBox in a PDCF file, and exactly one per each protected track.

```
aligned(8) class OMADRMKMSBox extends FullBox('odkm', version, 0) {
  OMADRMCommonHeaders Headers;  // Common headers box
  OMADRMAUFormatBox    AUFormat;  //optional
}
```

Table 13: OMA DRM Headers in PDCF

<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Headers</td>
<td>OMADRMCommonHeaders</td>
<td>OMA DRM Common headers as defined in 5.2.1.</td>
</tr>
</tbody>
</table>

Contained in the OMADRMKMSBox there MUST be:
- one OMADRMCommonHeaders box. The common headers box is exactly as defined in section 5.2.1

There MAY be:
- one OMADRMAUFormatBox, as the second sub-box..

7.1.3.1 Common Headers

The Common headers box is exactly the same as defined in section 5.2.1.

7.1.3.2 Access Unit Format Box

The OMADRMAUFormatBox is used to indicate the format of the OMADRMAUHeader which is placed on media access units.

```
aligned(8) class OMADRMAUFormatBox extends FullBox('odaf', 0, 0) {
```
bit(1) SelectiveEncryption;    // Must be one  
bite(7) reserved;     // Must be zero  
unsigned int(8) KeyIndicatorLength;   // Must be zero  
unsigned int(8) IVLength;  

Table 14 : OMA DRM Headers in PDCF

Where

SelectiveEncryption: Describes the use of Selective Encryption (refer [TS26.234] Annex K). This bit should be set to 1 in this version of the specification.

IVLength: Describes the size of the initialization vector in bytes. This length should be consistent with the algorithms used and indicated in table 1.

KeyIndicatorLength: Describes the size of the key indicator in bytes. In this version of the specification, the value of KeyIndicatorLength is 0.

In case the OMADRMAUFormatBox is omitted the default values for the fields are:
SelectiveEncryption: 1 (enabled)
KeyIndicatorLength: 0
IVLength: default value is as per the encryption mode (see section 5.2.12)

7.1.4 Access Unit Format Header

The Access Unit Format specifies the format for each access unit protected by OMA DRM. A media file format specifies the layout of the media data as samples, but the encryption/decryption process requires additional information carried in each access unit. The additional information is dependent on the DRM key management used. OMA DRM specifies its own access unit header, which MUST precede the codec-specific sample data in each access unit.

Table 15: PDCF Access Unit Format

<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>EncryptedAU</td>
<td>bit(1)</td>
<td>Encryption Indicator for the access unit.</td>
</tr>
<tr>
<td>IV</td>
<td>unsigned int(8 * IVLength)</td>
<td>IV data</td>
</tr>
<tr>
<td>KeyIndicator</td>
<td>unsigned int(8 * KeyIndicatorLength)</td>
<td>In this version of the specification, this field is not present as KeyIndicatorLength is zero...</td>
</tr>
</tbody>
</table>

Table 16: EncryptedAU Indicator values

<table>
<thead>
<tr>
<th>EncryptedAU</th>
<th>Value</th>
<th>Semantics</th>
</tr>
</thead>
</table>
When encrypting PDCF Content, the OMA DRM Access unit is encrypted. A playing Device uses the header information for decryption purposes and is able to extract the actual sample(s).

7.2 PDCF Streaming format (INFORMATIVE)

This section and its subsections are informative. This section describes how OMA DRM is applied to streaming content, especially in conjunction with a streaming service such as the 3GPP Packet switched Streaming Service (PSS) [TS26.234] or the 3GPP2 MSS [C.S0045].

Streaming DRM Content is leveraging the PDCF file format and widely deployed standard streaming protocols. DRM Content is transferred over a real-time streaming protocol as encrypted packets, which include the original payload. The encrypted payload wrapper format MAY be used in any streaming service using RTSP streaming [RFC2326], SDP signalling [RFC2327] and RTP transport [RFC3550].

Supporting the PDCF streaming is OPTIONAL, even if PDCF format is supported. A multimedia streaming session MAY consist of protected PDCF tracks and unprotected tracks.

Streaming protected tracks is signalled through SDP parameters, using information contained in the sample format entries of the PDCF file. A streaming server derives network packets from a hint track in the media file.

The streamable PDCF profiles are defined by each service supporting OMA DRM. In conjunction with the streamable file format, an end-to-end streaming service such as [TS26.234] or [C.S0045] MUST specify the RTP payload format used and mechanisms for signalling OMA DRM and encryption parameters. This specification defines the OMA DRM parameters that MUST be signalled in a streaming session.

7.2.1 RTP Payload

The RTP payload format consists of two parts: the payload wrapper and the actual media payload. The media payload (e.g. H.263 video) is packetized according to the appropriate standard, encrypted as required, and stored as packets in the PDCF file. The encrypted payload wrapper includes a header with additional signalling information, such as Selective Encryption indicator and initial vector for the packet. With this mechanism, one encrypted payload specification is used to protect any standard RTP payload. Also a benefit of the wrapper format is that the DRM system is fully functional in networks supporting basic RTP profiles, and thus not placing requirements on existing network configurations.

7.2.2 Session signalling

For PDCF streaming, the session descriptors (SDP files) MUST include information about the wrapper payload. The format parameters for the wrapper format are used to signal e.g. DRM Key Management Specific parameters and Encryption Parameters.

Each streaming service supporting PDCF streaming must allocate space for signalling OMA DRM Key Management Specific headers. In the SDP Encryption Parameters, PDCF streaming MUST support the AES 128 cipher in counter mode. If the Selective Encryption feature is disabled for a track, the Device MUST discard all packets belonging to this track where the encryption indicator is ‘false’ (unencrypted).

The Key Management Specific parameters MUST include the mandatory OMA DRM headers, as name value pairs. These parameters MUST be derived from the key management box in PDCF.

Table 17: Required OMA DRM specific parameters

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>ContentID</td>
<td>ContentID for the protected track</td>
</tr>
<tr>
<td>RightsIssuerURL</td>
<td>The RightsIssuerURL for fetching Rights</td>
</tr>
</tbody>
</table>
Other headers MAY be added to the key management specific parameters, and a consuming Device MUST pass them to the DRM Agent. The DRM Agent will then act accordingly and acquire Rights for the stream as appropriate. The semantics of the headers are the same as the common headers defined in section 5.2.
Appendix A. Static Conformance Requirements (Normative)

The notation used in this appendix is specified in [IOPPROC].

### A.1 Client Conformance Requirements

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
<th>Status</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRM_DCF-CLI-2</td>
<td>PDCF support</td>
<td>7</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>DRM_DCF-CLI-3</td>
<td>AES128CBC encryption algorithm</td>
<td>5.2.1.2</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>DRM_DCF-CLI-4</td>
<td>AES128CTR mode encryption algorithm</td>
<td>5.2.1.2</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>DRM_DCF-CLI-5</td>
<td>Ignore unsupported boxes</td>
<td>5.1</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>DRM_DCF-CLI-6</td>
<td>Common headers</td>
<td>5.2.1</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>DRM_DCF-CLI-7</td>
<td>Textual headers</td>
<td>5.2.2</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>DRM_DCF-CLI-8</td>
<td>GroupID</td>
<td>5.2.3.1</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>DRM_DCF-CLI-9</td>
<td>Mutable DRM Information box</td>
<td>5.2.4</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>DRM_DCF-CLI-10</td>
<td>Transaction Tracking box</td>
<td>5.2.4.1</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>DRM_DCF-CLI-11</td>
<td>Rights Object box</td>
<td>5.2.4.2</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>DRM_DCF-CLI-12</td>
<td>ISO format constraints</td>
<td>6.2.1</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>DRM_DCF-CLI-13</td>
<td>FullBox version</td>
<td>6.2.1</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>DRM_DCF-CLI-14</td>
<td>DCF header</td>
<td>6.2.2</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>DRM_DCF-CLI-15</td>
<td>OMA DRM container box</td>
<td>6.3.1</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>DRM_DCF-CLI-16</td>
<td>Discrete headers box</td>
<td>6.3.2</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>DRM_DCF-CLI-17</td>
<td>User-Data box</td>
<td>6.3.2.3</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>DRM_DCF-CLI-18</td>
<td>Content Object box</td>
<td>6.3.3</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>DRM_DCF-CLI-19</td>
<td>Multipart DCF</td>
<td>6.4</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>DRM_DCF-CLI-20</td>
<td>Extension boxes</td>
<td>6.3.4</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>DRM_DCF-CLI-21</td>
<td>UTF-8 character encoding for 3GPP asset information</td>
<td>6.3.2.3</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>DRM_DCF-CLI-22</td>
<td>UTF-16 character encoding for 3GPP asset information</td>
<td>6.3.2.3</td>
<td>O</td>
<td></td>
</tr>
</tbody>
</table>

### A.2 Client Conformance Requirements For The PDCF Format

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
<th>Status</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRM_DCF-CLI-23</td>
<td>3GPP conformance</td>
<td>7</td>
<td>O</td>
<td>Conform to [TS26.244]</td>
</tr>
<tr>
<td>DRM_DCF-CLI-24</td>
<td>3GPP2 conformance</td>
<td>7</td>
<td>O</td>
<td>Conform to [C.S0050]</td>
</tr>
<tr>
<td>DRM_DCF-CLI-25</td>
<td>OMA DRM key management</td>
<td>7.1.3</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>DRM_DCF-CLI-26</td>
<td>OMA DRM scheme</td>
<td>7.1.1</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>DRM_DCF-CLI-27</td>
<td>Common headers</td>
<td>5.2.1</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>DRM_DCF-CLI-28</td>
<td>AES128CTR mode encryption algorithm</td>
<td>5.2.1.2</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>DRM_DCF-CLI-29</td>
<td>Textual headers</td>
<td>5.2.2</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>DRM_DCF-CLI-30</td>
<td>GroupID</td>
<td>5.2.3.1</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>Item</td>
<td>Function</td>
<td>Reference</td>
<td>Status</td>
<td>Requirement</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------</td>
<td>-----------</td>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>DRM_DCF-CLI-31</td>
<td>Mutable DRM Information box</td>
<td>5.2.4</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>DRM_DCF-CLI-32</td>
<td>Transaction Tracking box</td>
<td>5.2.4.1</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>DRM_DCF-CLI-33</td>
<td>Rights Object box</td>
<td>5.2.4.2</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>DRM_DCF-CLI-34</td>
<td>OMA DRM access unit format</td>
<td>7.1.4</td>
<td>O</td>
<td></td>
</tr>
</tbody>
</table>
Appendix B. Reserved Numbers (Informative)

This Appendix lists common 4CC constants used in DCF and PDCF formats. The tables list only 4CC constants specified by OMA.

Table 18: Reserved identifier constants in the DCF format

<table>
<thead>
<tr>
<th>4CC</th>
<th>Reference</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>'ohdr'</td>
<td>5.2.1</td>
<td>Common headers box</td>
</tr>
<tr>
<td>'mdri'</td>
<td>5.2.4</td>
<td>Mutable DRM Information box</td>
</tr>
<tr>
<td>'grpi'</td>
<td>5.2.3.1</td>
<td>Group ID box</td>
</tr>
<tr>
<td>'odtt'</td>
<td>5.2.4.1</td>
<td>Transaction Tracking box</td>
</tr>
<tr>
<td>'odrb'</td>
<td>5.2.4.2</td>
<td>Rights Object box</td>
</tr>
<tr>
<td>'odcf'</td>
<td>6.2.2</td>
<td>File brand</td>
</tr>
<tr>
<td>'odrm'</td>
<td>6.3.1</td>
<td>OMA DRM Container box</td>
</tr>
<tr>
<td>'odhe'</td>
<td>6.3.2</td>
<td>Headers box for the Discrete Media profile box</td>
</tr>
<tr>
<td>'icnu'</td>
<td>6.3.2.3.5</td>
<td>Icon URI</td>
</tr>
<tr>
<td>'infu'</td>
<td>6.3.2.3.6</td>
<td>Info URL</td>
</tr>
<tr>
<td>'odda'</td>
<td>6.3.3</td>
<td>Content Object box</td>
</tr>
</tbody>
</table>

Table 19: Reserved OMA DRM specific identifier constants in the PDCF format

<table>
<thead>
<tr>
<th>4CC</th>
<th>Reference</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>'grpi'</td>
<td>5.2.3.1</td>
<td>Group ID box</td>
</tr>
<tr>
<td>'mdri'</td>
<td>5.2.4</td>
<td>Mutable DRM Information box</td>
</tr>
<tr>
<td>'odtt'</td>
<td>5.2.4.1</td>
<td>Transaction Tracking box</td>
</tr>
<tr>
<td>'odrb'</td>
<td>5.2.4.2</td>
<td>Rights Object box</td>
</tr>
<tr>
<td>'odkm'</td>
<td>7.1.2, 7.1.3</td>
<td>OMA DRM scheme type, OMA DRM scheme information box identifier</td>
</tr>
<tr>
<td>'ohdr'</td>
<td>7.1.3.1</td>
<td>Common headers box</td>
</tr>
<tr>
<td>'odaf'</td>
<td>7.1.3.2</td>
<td>Access Unit Format box</td>
</tr>
</tbody>
</table>
## Appendix C. Change History

(Informative)

### C.1 Approved Version History

<table>
<thead>
<tr>
<th>Reference</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OMA-TS-DRM-DCF-V2_0</td>
<td>03 Mar 2006</td>
<td>Status changed to Approved by TP TP Doc ref# OMA-TP-2006-0084R02-INP_DRM_V2_0_for_final_approval</td>
</tr>
<tr>
<td>24 Oct 2007</td>
<td></td>
<td>Updated with the current TS template</td>
</tr>
<tr>
<td>18 Jan 2008</td>
<td></td>
<td>General editorial clean-up Updated to the 2008 template</td>
</tr>
<tr>
<td>Approved Version</td>
<td>26 Feb 2008</td>
<td>Status changed to Approved by TP TP Doc ref# OMA-TP-2008-0082-INP_Digital_Rights_Management_V2_0_1_ERP_for_Notification.zip</td>
</tr>
</tbody>
</table>