Rights Expression Language Version 1.0

Version 13-September-2002

Open Mobile Alliance
OMA-Download-DRMREL-v1_0-20020913-C

.

A list of errata and updates to this document is available from the Open Mobile Alliance™ Web site, http://www.openmobilealliance.org/, in the form of SIN documents, which are subject to revision or removal without notice.

This document is available online in PDF format at http://www.openmobilealliance.org/.

Known problems associated with this document are published at http://www.openmobilealliance.org/.

Comments regarding this document can be submitted to the Open Mobile Alliance™ in the manner published at http://www.openmobilealliance.org/technical.htm

Document History	
OMA-Download-DRMREL-v1_0-20020913-C	Current
OMA-Download-DRMREL-v1_0-20020913-p	Proposed

Contents

1. Scope

Open Mobile Alliance (OMA) Wireless Application Protocol (WAP) is a result of continuous work to define an industry-wide specification for developing applications that operate over wireless communication networks. The scope for the Open Mobile Alliance is to define a set of specifications to be used by service applications. The wireless market is growing very quickly and reaching new customers and services. To enable operators and manufacturers to meet the challenges in advanced services, differentiation, and fast/flexible service creation, WAP defines a set of protocols in transport, session, and application layers. For additional information on the WAP architecture, refer to [WAPARCH].

The scope of OMA "*Digital Rights Management*" [DRM] is to enable the controlled consumption of digital media objects by allowing content providers to express usage rights, e.g., the ability to preview DRM content, to prevent downloaded DRM content from being illegally forwarded (copied) to other users, and to enable super distribution of DRM content. The defined technology is an initial DRM system that can be extended into a more comprehensive and secure DRM system.

The scope for this specification is to define the rights expression language used to describe the rights governing the usage of DRM content.

It addresses requirements such as enabling preview, i.e., test-driving, of content, possibly prior to purchasing, expressing a range of different permissions and constraints, and optimisation of rights objects delivered over constrained bearers. It provides a concise mechanism for expressing rights over DRM content. It is independent of the content being distributed, the mechanism used for distributing the content, and the billing mechanism used to handle the payments.

The OMA "*Digital Rights Management*" specification defines the context in which the DRM Rights Expression Language is used.

2. References

2.1. Normative References

[CREQ] "Specification of WAP Conformance Requirements". WAP Forum™. WAP-221-CREQ. http://www.openmobilealliance.org/

[ISO8601] "Representations of dates and times". ISO (International Organization for Standardization). http://www.iso.ch/markete/8601.pdf

[ODRL] "Open Digital Rights Language (ODRL)". Version 1.1. 8 August 2002. http://odrl.net/1.1/ODRL-11.pdf

[RFC2119] "Key words for use in RFCs to Indicate Requirement Levels". S. Bradner. March 1997. http://www.ietf.org/rfc/rfc2119.txt

[RFC2396] "Uniform Resource Identifiers (URI): Generic Syntax". T. Berners-Lee, R. Fielding, L. Masinter. August 1998. ftp://ftp.isi.edu/in-notes/rfc2396.txt

[WBXML] "Binary XML Content Format Specification". WAP Forum™. WAP-192-WBXML. http://www.openmobilealliance.org/

[XML] "Extensible Markup Language (XML) 1.0 (Second Edition)". W3C Recommendation 6 October 2000. http://www.w3c.org/TR/2000/REC-xml-20001006/

[XMLSchema] "XML Schema Part 2: Datatypes". W3C Recommendation 2 May 2001. http://www.w3.org/TR/xmlschema-2/

2.2. Informative References

[WAPARCH] "WAP Architecture". WAP Forum™. WAP-210-WAPArch. http://www.openmobilealliance.org/

[DRM] "Digital Rights Management". Open Mobile Alliance™. OMA-Download-DRM-v1_0. http://www.openmobilealliance.org/

[DRMCF] "DRM Content Format". Open Mobile Alliance™. OMA-Download-DRMCF-v1_0. http://www.openmobilealliance.org/

[XMLENC] "XML Encryption Syntax and Processing". W3C Candidate Recommendation 04 March 2002. http://www.w3.org/TR/2002/CR-xmlenc-core-20020304/

[XMLSIG] "XML Signature Syntax and Processing". W3C Recommendation 12 February 2002. http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/

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

Asset	Content governed by rights. See DRM content.
Combined delivery	Delivery of the rights object and content together in a single message. See DRM message.
Composite object	A media object that contains one or more media objects by means of inclusion.
Consuming device	A mobile device consuming DRM content e.g. DRM messages, ZIP file.
Content	A media object.
DRM agent	A user agent in the device that enforces the rights and controls the consumption of DRM content on the device.
DRM content	Content that is consumed according to rights. DRM content may be in encrypted DRM Content Format or in plaintext delivered inside a DRM message.
DRM message	A message containing a media object and an optional rights object. Media objects received inside a DRM message must not leave the device. The optional rights object defines additional consumption rules for the media object.
Forward-lock	A special case of combined delivery method where the DRM message includes only the media object and not a rights object at all. A set of default rights applies for the media object.
Media object	A digital resource e.g. a ringing tone, screen saver, Java game or any other digital resource, or a composite object.
Rights	Permissions and constraints defining under which circumstances access is granted to DRM content.
Rights object	An instance of rights.
Separate delivery	Delivery of rights object and content via separate transports.
Superdistribution	A mechanism that (1) allows the end user to redistribute the encrypted DRM content to other end users through potentially insecure channels and (2) enables the receipients to use the rights refresh mechanism to obtain initial rights for the superdistributed DRM content.

3.3. Abbreviations

AES	Advanced Encryption Standard
CEK	Content Encryption Key
DRM	Digital Rights Management
MIME	Multipurpose Internet Mail Extensions
OMA	Open Mobile Alliance
ODRL	Open Digital Rights Language
REL	Rights Expression Language
SMS	Short Message Service
WAP	Wireless Application Protocol
WBXML	WAP Binary XML
XML	Extensible Markup Language

4. Introduction

Digital Rights Management [DRM] defines the mechanisms how to deliver DRM content and rights objects to a consuming device. Rights are used to specify the access a consuming device is granted to DRM content. The Rights Expression Language (REL) defined in this document specifies the syntax and semantics of rights governing the usage of DRM content based on the Open Digital Rights Language [ODRL].

DRM content is consumed according to the specified rights. Therefore, the value is in the rights and not in the content itself. Thus, rights objects must not leave the consuming device. Furthermore, the consuming device must not modify the rights objects.

For reasons of simplicity and efficiency, one rights object references one piece of content at a time. This cardinality between content and rights objects can easily be extended in future versions of this specification if a more fine-grained specification of content-rights association becomes necessary. Rights governing a composite object, e.g., a theme consisting of a ringing tone and a logo, apply to the individual components of the composite content. Note that if the individual components can be referenced separately, multiple rights objects can be used to specify rights for each of the individual components directly.

4.1. Goals

The goal of this specification is to define a REL taking into account the special requirements and characteristics of the mobile domain to express consumption rights over DRM content. Some of the specific goals include

- Light-weight and simple way of expressing rights
- Minimal set of permissions and constraints
- Quick and easy to implement and deploy in short time to market
- Lowering the entrance barrier for content providers and other players to adopt DRM technologies
- Optimised expression of rights for delivery over constrained bearers
- Suitable for specifying rights independently of the content type
- Suitable for specifying rights independently of the transport mechanism
- Suitable for specifying rights to encrypted and unencrypted content
- Enable specification of right to preview, i.e., test-drive, DRM content enabling users to experience the content first hand, possibly prior to purchasing it
- Enable specification of constraints to restrict permissions to the number of times content can be accessed, and time limits and intervals during which content can be accessed.

4.2. Non-goals

OMA Digital Rights Management [DRM] defines a simple DRM system. Therefore, the REL only comprises features that are required for the simple DRM system freeing the REL from unnecessary complexity. In particular, the following are not goals of the REL:

- To govern the distribution of DRM content is not a goal of the REL. If DRM content is encrypted it is useless without the corresponding rights, and thus there is no need for the REL to explicitly govern distribution. If DRM content is in plain, distributing it would be highly irresponsible. Thus, also in this case, the REL does not need to explicitly govern distribution.
- To govern device management permissions such as ‘install’, ‘uninstall’, ‘delete’, etc. Freeing memory capacity on the device is an intrinsic right that every user has to his/her device.

5. Structure

This section describes the structure of the rights expression language. The REL is defined as a mobile profile of ODRL v1.1 [ODRL]. Rights are the collection of permissions and constraints defining under which circumstances access is granted to DRM content. The structure of the rights expression language enables the following functionality:

1. Metadata such as version and content ID
2. The actual rights specification consisting of
    a. Linking to and providing protection information for the content, and
    b. Specification of usage rights and constraints

Models are used to group rights elements according to their functionality, and thus enable concise definition of elements and their semantics. The following models are used throughout this specification:

- Foundation model
- Agreement model
- Context model
- Permission model
- Constraint model
- Security model

The rights expression language is defined as a mobile profile, i.e., a subset, of ODRL. Section 5.7 specifies how to handle ODRL models and elements that are not used in this specification.

5.1. Foundation Model

The foundation model constitutes the basis for rights. It consists of the <context> and <rights> elements bringing together Meta information and agreement information. The foundation model serves as the starting point for incorporating the agreement model and the context model.

5.1.1. Element <rights>

Element	<!ELEMENT o-ex:rights (o-ex:context, o-ex:agreement)>
Semantics	The <rights> element is the root element of all rights objects defined according to this specification. It contains the mandatory <context> and <agreement> elements linking assets to corresponding permissions.

5.2. Agreement Model

The agreement model expresses the permissions that are granted over an asset. It consists of the <agreement> element connecting a set of permissions with the corresponding asset. The asset element itself is also part of the agreement model. The agreement model incorporates the permission model and the security model.

5.2.1. Element <agreement>

Element	<!ELEMENT o-ex:agreement (o-ex:asset, o-ex:permission)>
Semantics	The <agreement> element specifies the permissions granted over the corresponding asset. It contains the mandatory <asset> and <permission> elements.

5.2.2. Element <asset>

Element	<!ELEMENT o-ex:asset (o-ex:context, ds:KeyInfo?)>
Semantics	The <asset> element specifies the identity of the content governed by the containing <agreement> element via the <context> child element. The optional <KeyInfo> element provides the functionality to access the encrypted content if granted the rights to do so. Note that the <KeyInfo> element is omitted and MUST be ignored by the DRM agent if the content is unencrypted.

5.3. Context Model

The context model provides Meta information about the rights. It augments the foundation model, the agreement model, and the constraint model by expressing additional information.

The <context> element is used in the <rights> element and in the <asset> element. As the model's name already indicates, the semantics of its child elements depend on the context in which it occurs in the rights object.

5.3.1. Element <context>

Element	<!ELEMENT o-ex:context (o-dd:version?, o-dd:uid?)>
Semantics	The <context> element contains the optional <version> and <uid> elements. As the name already indicates, it provides context sensitive information for use within the context of its parent element. The semantics of its child elements depend on the parent element in which the <context> element is used. These are different if the <context> element is a child element of the <rights> element from when the <context> element is a child element of the <asset> element. Please see the corresponding descriptions of the individual child elements.

5.3.2. Element <version>

Element	<!ELEMENT o-dd:version (#PCDATA)>
Semantics	The <version> element SHOULD only be used if its parent <context> element is included in the <rights> element. The <version> element then specifies the version of the rights object. For this specification its content MUST be "1.0" (without quotes).

5.3.3. Element <uid>

Element	<!ELEMENT o-dd:uid (#PCDATA)>
Semantics	If its parent <context> element is included in the <asset> element, the <uid> element specifies the content identifier of the corresponding DRM content. It contains the ContentURI value of the DCF [DRMCF] in case of encrypted content (separate delivery) or the Content-Id header of the referenced media object inside the DRM message in the case of unencrypted content (combined delivery) [DRM]. The format used for the value MUST conform to [RFC2396].

5.4. Permission Model

The permission model augments the agreement model. It facilitates the expression of permissions over assets by specifying the access granted to a device. The permission model incorporates the constraint model allowing fine-grained consumption control of content.

The set of permissions comprises <play>, <display>, <execute>, and <print>. Access to the content MUST only be granted according to the permissions explicitly specified by corresponding rights.

Note that the REL only specifies consumption rights and not management rights, e.g., install, uninstall, delete, or distribution rights, e.g., forward, copy. This is made possible by the separation of content and rights objects (although content and rights objects may be delivered together) freeing the REL from unnecessary complexity and overhead. Content, in encrypted and unencrypted form, can be stored; however, it can only be accessed if corresponding rights are available.

Similarly, encrypted content can be super-distributed without unnecessarily complicating the REL; no separate distribution permissions are necessary, since encrypted content without the decryption key is of no value.

Unknown or unsupported permission elements MUST be ignored and alternative, not explicitly specified rights to access content MUST NOT be granted instead. Other known and supported permissions defined by the same rights object MUST remain unaffected and access MUST be granted according to those. A permission that is not granted due to unknown or unsupported constraints (section 5.5) MUST NOT affect the granting of other permissions.

5.4.1. Element <permission>

Element	<!ELEMENT o-ex:permission (o-dd:play?, o-dd:display?, o-dd:execute?, o-dd:print?)>
Semantics	The <permission> element contains a set of optional permissions specifying the rights over a piece of content. It contains the optional <play>, <display>, <execute>, and <print> elements.

5.4.2. Element <play>

Element	<!ELEMENT o-dd:play (o-ex:constraint?)>
Semantics	The <play> element grants play rights over an asset. It contains an optional <constraint> element. If the <constraint> element is specified the DRM agent MUST grant play rights according to the <constraint> child element. If no <constraint> element is specified, the DRM agent MUST grant unlimited play rights. The <play> element has the semantics of rendering the DRM content into audio/video form, for example, audio/midi, video/quicktime. The DRM agent MUST NOT grant access according to <play> to content that cannot be rendered in this way. Note that the DRM agent MUST NOT grant access to game content, e.g., Java™ games, based on the <play> permission. In order to specify rights for Java™ games, the <execute> element MUST be utilized instead (section 5.4.4).

5.4.3. Element <display>

Element	<!ELEMENT o-dd:display (o-ex:constraint?)>
Semantics	The <display> element grants display rights over an asset. It contains an optional <constraint> element. If the <constraint> element is specified the DRM agent MUST grant display rights according to the <constraint> child element. If no <constraint> element is specified, the DRM agent MUST grant unlimited display rights. The <display> element has the semantics of rendering the DRM content onto a visual device, for example, image/gif or image/jpeg. The DRM agent MUST NOT grant access according to <display> to content that cannot be rendered in this way.

5.4.4. Element <execute>

Element	<!ELEMENT o-dd:execute (o-ex:constraint?)>
Semantics	The <execute> element grants execution rights over an asset. It contains an optional <constraint> element. If the <constraint> element is specified the DRM agent MUST grant execution rights according to the <constraint> child element. If no <constraint> element is specified, the DRM agent MUST grant unlimited execution rights. The <execute> element has the semantics of executing, i.e., invoking, DRM content, e.g., Java™ games or other applications. Thus, the DRM agent MUST NOT grant access according to <execute> to content that cannot be rendered in this way.

5.4.5. Element <print>

Element	<!ELEMENT o-dd:print (o-ex:constraint?)>
Semantics	The <print> element grants print rights over an asset. It contains an optional <constraint> element. If the <print> element is specified the DRM agent MUST grant print rights according to the <constraint> child element. If no <constraint> element is specified, the DRM agent MUST grant unlimited print rights. The <print> element has the semantics of printing, i.e., creating a hardcopy of, the DRM content, for example, image/jpeg. The DRM agent MUST NOT grant access according to <print> to content that cannot be rendered in this way.

5.5. Constraint Model

The constraint model enhances the permission model by providing fine-grained consumption control of content.

Constraints are associated with one permission element at a time. For a permission to be granted all its constraints MUST be fulfilled. If a constraint is not understood or cannot be enforced by the consuming device the parent permission is invalid and MUST NOT be granted.

5.5.1. Element <constraint>

Element	<!ELEMENT o-ex:constraint (o-dd:count?, o-dd:datetime?, o-dd:interval?)>
Semantics	The <constraint> element is the top most element in the constraint model. It contains the optional <count>, <datetime>, and <interval> elements.

5.5.2. Element <count>

Element	<!ELEMENT o-dd:count (#PCDATA)>
Semantics	The <count> element specifies the number of times a permission may be granted over an asset. It contains a positive integer value. The DRM agent MUST NOT grant access to the DRM content more often than specified by the value of the child <fixed> element. Similarly, the DRM agent MUST NOT grant access to the DRM content if the value of the child <fixed> element is non-positive.

5.5.3. Element <datetime>

Element	<!ELEMENT o-dd:datetime (o-dd:start?, o-dd:end?)>

Semantics	The <datetime> element specifies the time range, respectively the time limit, for a containing permission. It contains the optional <start> and <end> elements. If the <start> element is present, its semantics are 'not before' the specified time/date. If the <end> element is present, its semantics are 'not after' the specified time/date. If both are present, the value of the <start> element MUST be smaller than the value of the <end> element. If the value of the <start> element is greater than the value of the <end> element then the DRM agent MUST NOT grant access to the DRM content according to the containing permission. If both are absent, the datetime element does not have a meaning and MUST be ignored. The DRM agent of a consuming device without a time source MUST NOT grant access to DRM content according to permissions containing the <datetime> element. Note that, strictly speaking, the effective and secure enforcement of the <datetime> element requires a secure time source that cannot be tampered with by the end user. Without a secure clock, an end user could easily obtain prolonged rights to content by setting the device's date and time at will.

5.5.3.1. Element <start>

Element	<!ELEMENT o-dd:start (#PCDATA)>
Semantics	The <start> element specifies the start time/date. Its semantics are 'not before'. The general format used for specifying time/date values is defined in [ISO8601]. To increase interoperability and facilitate ease of implementation values MUST conform to a single lexical representation defined in section 3.2.7 of [XMLSchema]. This lexical representation is the extended format CCYY-MM-DDThh:mm:ss where CC denotes the century, YY denotes the year, MM denotes the month, DD denotes the day, T is the date/time separator, and hh, mm, ss represent the hour, minute, and second respectively. For example, 2002-12-31T23:59:59 represents December 31st, 2002, 23:59:59 local time. The DRM agent MUST NOT grant access to the DRM content before the time/date specified by the value of the <start> element.

5.5.3.2. Element <end>

Element	<!ELEMENT o-dd:end (#PCDATA)>
Semantics	The <end> element specifies the end time/date. Its semantics are 'not after'. The general format used for specifying time/date values is defined in [ISO8601]. To increase interoperability and facilitate ease of implementation values MUST conform to a single lexical representation defined in section 3.2.7 of [XMLSchema]. This lexical representation is the extended format CCYY-MM-DDThh:mm:ss where CC denotes the century, YY denotes the year, MM denotes the month, DD denotes the day, T is the date/time separator, and hh, mm, ss represent the hour, minute, and second respectively. For example, 2002-12-31T23:59:59 represents December 31st, 2002, 23:59:59 local time. The DRM agent MUST NOT grant access to the DRM content after the time/date specified by the value of the <end> element.

5.5.4. Element <interval>

Element	<!ELEMENT o-dd:interval (#PCDATA)>
Semantics	The <interval> element specifies a recurring period of time during which the rights can be exercised over the DRM content. The general format used for specifying interval values is defined in [ISO8601]. To increase interoperability and facilitate ease of implementation values MUST conform to a single lexical representation defined in section 3.2.6 of [XMLSchema]. This lexical representation is PnYnMnDTnHnMnS or any reduced version thereof as specified in [XMLSchema]. For example, P2Y10M15DT10H30M20S represents a duration of 2 years, 10 months, 15 days, 10 hours, 30 minutes and 20 seconds. The DRM agent MUST NOT grant access to the DRM content after the period specified by the value of the <interval> element has elapsed. The DRM agent of a consuming device without a time source MUST NOT grant access to DRM content according to permissions containing the <interval> element. Note that, strictly speaking, the effective and secure enforcement of the <interval> element requires a secure time source that cannot be tampered with by the end user. Without a secure clock, an end user could easily obtain prolonged rights to content by setting the device's date and time at will.

5.6. Security Model

Often, security is not incorporated into the system design from the beginning. Rather, it is frequently added at a later point in time when it becomes evident that the underlying business model is at risk without providing adequate security. Not uncommonly, by then the system's design has progressed to an extent that changes to the architecture required for increasing the security are no longer possible.

Security constitutes an important part of a DRM system. Even if a particular DRM system is not designed to provide the technically highest possible degree of security (because of other factors such as business models, cost of increased security vs. value of content, etc.) security must be accounted for in the REL used to express rights over DRM content in this system. In order not to make the REL the bottleneck for future improvements in security and prevent stepping up

the security of the system by the design of the REL, the functionality required for a very high level of security is described in this specification. Only the minimal set of security elements required to enable the functionality described in [DRM], however, are normative (section 5.6.2) while those parts outlining the security features beyond the scope of [DRM] are informative (sections 5.6.1 and 5.6.3).

The security model enhances the agreement model. It is designed to

1. Enforce the integrity of rights
2. Ensure the controlled consumption of DRM content
3. Enforce the integrity of the association between rights and DRM content

The ODRL security model, which forms the basis for the security model of this specification, is based on [XMLENC] and [XMLSIG].

Note that only the controlled consumption of DRM content is a normative part of this specification. Ensuring the integrity of rights and of the association of rights and DRM content is left to future versions of this specification.

5.6.1. Rights Integrity (Informative)

Integrity protection prevents illegitimately modifying the rights specified over DRM content, including but not limited to, adding, deleting, and modifying permissions and constraints over an asset, references to the asset itself, and meta information included in the rights.

This specification can easily be extended utilizing functionality from [XMLSIG] to ensure the integrity of rights. Since it is not part of the functionality required by [DRM], the description of the corresponding elements and their functionality is not part of this specification.

5.6.2. Content Confidentiality

Protecting content confidentiality is an essential part of enforcing consumption control of DRM content. Enabling an authorized party to consume content is similar to granting this party access to the confidential content. In other words, a party authorized to consume content is let into the exclusive circle of parties deemed trustworthy enough to access the protected content.

This concept is realized in [DRM] by i) encrypting the DRM content [DRMCF], and ii) sharing the key required to decrypt the DRM content only with those parties that are authorized to consume the content.

Content is encrypted using a symmetric algorithm (AES), i.e., the key used for decryption can be derived from the key used for encryption. Thus, henceforth, the key will be referred to as *content encryption key*, or short *CEK*. Encrypting the content defers content confidentiality to controlling the confidentiality of the CEK. Now, the security of the DRM system relies on the control of the CEK that must be kept secret from all unauthorized parties.

[DRM] specifies the means of making the key necessary for content decryption available to authorized parties. The CEK is not encrypted and thus its confidentiality is dependent on the delivery mechanism [DRM].

The security elements used to achieve the level of security required by [DRM] are described next.

5.6.2.1. Element <KeyInfo>

Element	<!ELEMENT ds:KeyInfo (ds:KeyValue)>
Semantics	The <KeyInfo> element is the starting point for all consumption control, i.e., content encryption, functionality. It contains the <KeyValue> element. The <KeyInfo> element associates the corresponding protection with the asset governed by the rights. Note that the <KeyInfo> element MUST NOT be included in the <asset> element if the corresponding DRM content is not encrypted [DRMCF].

5.6.2.2. Element <KeyValue>

Element	<!ELEMENT ds:KeyValue (#PCDATA)>
Semantics	The <KeyValue> element contains the content encryption key required for content consumption in plain. The content of this element is base64 encoded. Note that the content of this element is in binary format when rights object is WBXML [WBXML] encoded (see section 7).

5.6.3. Rights-Content Association Integrity (Informative)

The ability to replace the DRM content governed by rights amounts to the ability of changing the rights itself. Thus, the association between rights and the corresponding DRM content must be integrity protected as much as the specified rights itself.

A reference to a piece of DRM content is established via the <uid> element contained in the <context> element of the <asset> element. Enforcing the integrity of the association between rights and DRM content is handled similarly to enforcing the integrity of rights (section 5.6.1), i.e., by signing rights objects utilizing functionality from [XMLSIG]. This prevents tampering with the content identifier in the rights object. It does not, however, prevent from modifying the corresponding identifier in the DRM content [DRMCF] itself.

It is possible to provide a way of securing the content-end of the rights-content association without having to sign the DRM content. Instead, a hash of the (encrypted) DRM content is included in the rights object. Since this hash value is part of the signed rights object, it is as safe from being tampered with, as is the <uid> element in the rights object referencing the content. The integrity of the content-end is guaranteed by the very characteristics of the hash itself: any modifications to the DRM content automatically invalidate the hash value inside the rights object. Note that including the hash of the content in the rights object only ensures the integrity of the rights-content association if the integrity of the rights is protected also, e.g., by signing them.

Since signing rights objects is not required for the DRM system [DRM], the description of elements ensuring the integrity of rights-content association are not described in this specification.

5.7. ODRL Compatibility

This specification defines a mobile profile of ODRL v1.1 [ODRL]. This specification takes precedence in case there is any divergence from [ODRL].

The DRM agent of a consuming device encountering any ODRL [ODRL] elements not defined within this specification MUST proceed as follows:

- Unsupported permissions MUST be ignored. Supported permissions MUST still be granted.
- Permissions containing one or more unsupported constraints MUST NOT be granted.
- Rights objects containing any unsupported <requirement> elements MUST NOT be granted.
- Rights objects containing any unsupported <condition> elements MUST NOT be granted.
- Unsupported <rightsholder> elements SHOULD be ignored.
- Unsupported <context> elements SHOULD be ignored.
- Unsupported <offer> elements SHOULD be ignored.
- Unsupported <party> elements within an <asset> element SHOULD be ignored.
- Unsupported <revoke> elements SHOULD be ignored.
- Unsupported elements of the ODRL security model SHOULD be ignored.

6. Syntax

Figure 1 depicts the syntax of the ODRL mobile profile using XML document type definition [XML]. The document type definition has the public identifier '-//OMA//DTD DRMREL 1.0//EN' and is located at 'http://www.openmobilealliance.org/DTD/drmrel10.dtd'. A consuming device MUST be able to parse all elements defined in this specification. Elements that are not defined in Figure 1 or not required according to corresponding static conformance requirements MUST be ignored.

```
<!ELEMENT o-ex:rights (o-ex:context, o-ex:agreement)>
<!ATTLIST o-ex:rights
      xmlns:o-ex CDATA #FIXED "http://odrl.net/1.1/ODRL-EX"
      xmlns:o-dd CDATA #FIXED "http://odrl.net/1.1/ODRL-DD"
      xmlns:ds CDATA #FIXED "http://www.w3.org/2000/09/xmldsig#/"
>
<!ELEMENT o-ex:context (o-dd:version?, o-dd:uid?)>
<!ELEMENT o-dd:version (#PCDATA)>
<!ELEMENT o-dd:uid (#PCDATA)>
<!ELEMENT o-ex:agreement (o-ex:asset, o-ex:permission)>
<!ELEMENT o-ex:asset (o-ex:context, ds:KeyInfo?)>
<!ELEMENT ds:KeyInfo (ds:KeyValue)>
<!ELEMENT ds:KeyValue (#PCDATA)>
<!ELEMENT o-ex:permission (o-dd:play?, o-dd:display?, o-dd:execute?, o-dd:print?)>
<!ELEMENT o-dd:play (o-ex:constraint?)>
<!ELEMENT o-dd:display (o-ex:constraint?)>
<!ELEMENT o-dd:execute (o-ex:constraint?)>
<!ELEMENT o-dd:print (o-ex:constraint?)>
<!ELEMENT o-ex:constraint (o-dd:count?, o-dd:datetime?, o-dd:interval?)>
<!ELEMENT o-dd:count (#PCDATA)>
<!ELEMENT o-dd:datetime (o-dd:start?, o-dd:end?)>
<!ELEMENT o-dd:start (#PCDATA)>
<!ELEMENT o-dd:end (#PCDATA)>
<!ELEMENT o-dd:interval (#PCDATA)>
```

Figure 1. Syntax.

7. WBXML Encoding

In addition to the textual XML representation, rights objects can be compacted using WBXML [WBXML] for transmission over constrained bearers, e.g., WAP Push over SMS. The DRM agent MUST support WBXML encoded rights objects.

Note that the encoding does not necessarily have to take place at the gateway but may also be performed at application level. WBXML yields up to 84% space savings compared to textual XML representation (section 8.3.2.2).

7.1. WBXML Encoding Rules

The following rules MUST be followed when WBXML encoding rights objects.

- WBXML version number 1.3 MUST be used (encoded as u_int8 value 0x03).
- The public identifier value "-//OMA//DTD DRMREL 1.0//EN" MUST be used (encoded as mb_u_int32 value 0x0E).
- The character set MUST be UTF-8 (encoded as mb_u_int32 value 0x6A).
- The content of the <KeyValue> element MUST be in binary format using the 'opaque' token.

7.2. Token Definitions

Element name	WBXML tag token (hex value)	Comment
o-ex:rights	05	
o-ex:context	06	
o-dd:version	07	
o-dd:uid	08	
o-ex:agreement	09	
o-ex:asset	0A	
ds:KeyInfo	0B	
ds:KeyValue	0C	Encoded in binary format, i.e., no base64 encoding
o-ex:permission	0D	
o-dd:play	0E	
o-dd:display	0F	
o-dd:execute	10	
o-dd:print	11	
o-ex:constraint	12	
o-dd:count	13	
o-dd:datetime	14	
o-dd:start	15	
o-dd:end	16	
o-dd:interval	17	

Attribute name	WBXML tag token (hex value)	Comment
xmlns:o-ex	05	
xmlns:o-dd	06	
xmlns:ds	07	

Attribute value	WBXML tag token (hex value)	Comment
http://odrl.net/1.1/ODRL-EX	85	ODRL Expression Language
http://odrl.net/1.1/ODRL-DD	86	ODRL Data Dictionary
http://www.w3.org/2000/09/xmldsig#/	87	XML Digital Signature

8. MIME Type

The MIME type for rights objects in textual XML representation MUST be

application/vnd.oma.drm.rights+xml

The MIME type for WBXML encoded rights objects MUST be

application/vnd.oma.drm.rights+wbxml

The file extension for textual rights objects MUST be ".dr" and the file extension for WBXML encoded rights objects MUST be ".drc".

Appendix A. Static Conformance Requirements (Normative)

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

8.1. Terminal Features

Item	Function	Reference	Status	Requirement
DRMREL-GEN-C-001	Separate delivery		O	DRMREL:MCF AND DRMREL-GEN-C-022 AND DRMREL-GEN-C-023 AND DRMREL-GEN-C-025
DRMREL-GEN-C-002	Syntax parsing	6	M	
DRMREL-GEN-C-003	<rights> element	5.1.1	M	
DRMREL-GEN-C-004	<agreement> element	5.2.1	M	
DRMREL-GEN-C-005	<asset> element	5.2.2	M	
DRMREL-GEN-C-006	<context> element	5.3.1	M	
DRMREL-GEN-C-007	<version> element	5.3.2	M	
DRMREL-GEN-C-008	<uid> element	5.3.2	M	
DRMREL-GEN-C-009	Unknown permissions	5.4	M	
DRMREL-GEN-C-010	<permission> element	5.4.1	M	
DRMREL-GEN-C-011	<play> element	5.4.2	M	
DRMREL-GEN-C-012	<display> element	5.4.3	M	
DRMREL-GEN-C-013	<execute> element	5.4.4	M	
DRMREL-GEN-C-014	<print> element	5.4.5	M	
DRMREL-GEN-C-015	Unknown constraints	5.5	M	
DRMREL-GEN-C-016	<constraint> element	5.5.1	M	
DRMREL-GEN-C-017	<count> element	5.5.2	M	
DRMREL-GEN-C-018	<datetime> element	5.5.3	O	DRMREL-GEN-C-019 AND DRMREL-GEN-C-020
DRMREL-GEN-C-019	<start> element	5.5.3.1	O	
DRMREL-GEN-C-020	<end> element	5.5.3.2	O	
DRMREL-GEN-C-021	<interval> element	5.5.4	O	
DRMREL-GEN-C-022	<KeyInfo> element	5.6.2.1	O	DRMREL-GEN-C-024
DRMREL-GEN-C-023	<KeyValue> element	5.6.2.2	O	
DRMREL-GEN-C-024	ODRL compatibility	5.7	M	
DRMREL-GEN-C-025	WBXML Encoding rules	7	O	
DRMREL-GEN-C-026	MIME type	8	M	

Appendix B. Change History (Informative)

Type of Change	Date	Section	Description
Class 0	13-September-2002		The initial version of this document.

Appendix C. Examples (Informative)

8.2. Combined Delivery of Rights and Content

This section contains two examples of rights objects for the combined delivery with content. The examples in sections 8.2.1 and 8.2.2 do not contain decryption keys, since the content is in plain.

No corresponding WBXML representations are shown as rights objects for combined delivery are in textual XML.

8.2.1. Play

The rights depicted in this example grant unconstrained permission to play the DRM content. Note that since content delivered in a DRM message must not be super-distributed, it also implements forward-lock functionality.

```
<o-ex:rights
    xmlns:o-ex="http://odrl.net/1.1/ODRL-EX"
    xmlns:o-dd="http://odrl.net/1.1/ODRL-DD"
>
  <o-ex:context>
    <o-dd:version>1.0</o-dd:version>
  </o-ex:context>
  <o-ex:agreement>
    <o-ex:asset>
      <o-ex:context>
        <o-dd:uid>cid:4567829547@foo.com</o-dd:uid>
      </o-ex:context>
    </o-ex:asset>
    <o-ex:permission>
      <o-dd:play/>
    </o-ex:permission>
  </o-ex:agreement>
</o-ex:rights>
```

8.2.2. Preview

The rights depicted in this example grant the right to display the DRM content once, thus implementing the functionality to test-drive, i.e., preview, content. Note that since content delivered in a DRM message must not be super-distributed, it also implements forward-lock functionality.

```
<o-ex:rights
    xmlns:o-ex="http://odrl.net/1.1/ODRL-EX"
    xmlns:o-dd="http://odrl.net/1.1/ODRL-DD"
>
  <o-ex:context>
    <o-dd:version>1.0</o-dd:version>
  </o-ex:context>
  <o-ex:agreement>
    <o-ex:asset>
      <o-ex:context>
        <o-dd:uid>cid:4567829547@foo.com</o-dd:uid>
      </o-ex:context>
```

```
    </o-ex:asset>
    <o-ex:permission>
      <o-dd:display>
        <o-ex:constraint>
          <o-dd:count>1</o-dd:count>
        </o-ex:constraint>
      </o-dd:display>
    </o-ex:permission>
  </o-ex:agreement>
</o-ex:rights>
```

Note that in this example preview functionality is implemented as displaying the content once. It is possible to implement any kind of preview through the use of appropriate constraints, e.g., limiting the time range during which content can be displayed through the use of the <datetime> and <interval> constraints (sections 5.5.3 and 5.5.4).

8.3. Separate Delivery of Rights and Content

This section contains two examples of rights objects for the separate delivery of content implementing the same functionality as the examples in sections 8.2.1 and 8.2.2. Both, the textual XML representations (sections 8.3.1.1 and 8.3.2.1) and the corresponding WBXML representations for delivery over constrained bearers (sections 8.3.1.2 and 8.3.2.2) are shown.

8.3.1. Play

The rights depicted in this example grant unconstrained permission to play the corresponding DRM content.

8.3.1.1. XML Representation

```
<o-ex:rights
    xmlns:o-ex="http://odrl.net/1.1/ODRL-EX"
    xmlns:o-dd="http://odrl.net/1.1/ODRL-DD"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#/"
>
  <o-ex:context>
    <o-dd:version>1.0</o-dd:version>
  </o-ex:context>
  <o-ex:agreement>
    <o-ex:asset>
      <o-ex:context>
        <o-dd:uid>cid:4567829547@foo.com</o-dd:uid>
      </o-ex:context>
      <ds:KeyInfo>
        <ds:KeyValue>Cei4QxjWo9Kg8D3pKgxw=</ds:KeyValue>
      </ds:KeyInfo>
    </o-ex:asset>
    <o-ex:permission>
      <o-dd:play/>
    </o-ex:permission>
  </o-ex:agreement>
</o-ex:rights>
```

Note that the CEK inside the <KeyValue> element is base64 encoded.

8.3.1.2. WBXML Encoding

Token stream (numbers in hexadecimal)	Description
03	WBXML version number – WBXML version 1.3
0E	Public identifier (-//OMA//DTD DRMREL 1.0//EN)
6A	Charset = UTF-8
00	String table length = 00 (empty string table)
C5	<o-ex:rights>
05	xmlns:o-ex=
85	"http://odrl.net/1.1/ODRL-EX" (attribute value)
06	xmlns:o-dd=
86	"http://odrl.net/1.1/ODRL-DD" (attribute value)
07	xmlns:ds=
87	http://www.w3.org/2000/09/xmldsig#/" (attribute value)
46	<o-ex:context>
47	<o-dd:version>
03	STR_I (inline string follows with a terminator)
"1.0", 00	"**1.0**" + string terminator
01	</o-dd:version>
01	</o-ex:context>
49	<o-ex:agreement>
4A	<o-ex:asset>
46	<o-ex:context>
48	<o-dd:uid>
03	STR_I (inline string follows with a terminator)
"cid:4567829547@foo.com", 00	"**cid:4567829547@foo.com**" + string terminator
01	</o-dd:uid>
01	</o-ex:context>
4B	<ds:KeyInfo>
4C	<ds:KeyValue>
C3	Opaque token
10	Length of opaque data (16 bytes = 0x10)
IeC8XrhM9lwfp4Hn	128 bit content encryption key (in binary form, i.e., no base64 encoding)
01	</ds:KeyValue>
01	</ds:KeyInfo>
01	</o-ex:asset>
4D	<o-ex:permission>
0E	<o-dd:play/>
01	</o-ex:permission>
01	</o-ex:agreement>
01	</o-ex:rights>

This example can be encoded in a total of 78 bytes compared to 466 bytes in the XML version. This is a saving of 83%.

Note that the CEK is in binary format (and not base64 encoded) for WBXML encoded rights objects.

8.3.2. Preview

The rights depicted in this example grant the right to display the corresponding content once, thus implementing the functionality to test-drive, i.e., preview, content.

Note that in this example preview functionality is implemented as displaying the content once. It is possible to implement any kind of preview through the use of appropriate constraints, e.g., limiting the time range during which content can be displayed through the use of the <datetime> and <interval> constraints (sections 5.5.3 and 5.5.4).

8.3.2.1. XML Representation

```
<o-ex:rights
    xmlns:o-ex="http://odrl.net/1.1/ODRL-EX"
    xmlns:o-dd="http://odrl.net/1.1/ODRL-DD"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#/"
>
  <o-ex:context>
    <o-dd:version>1.0</o-dd:version>
  </o-ex:context>
  <o-ex:agreement>
    <o-ex:asset>
      <o-ex:context>
        <o-dd:uid>cid:4567829547@foo.com</o-dd:uid>
      </o-ex:context>
      <ds:KeyInfo>
        <ds:KeyValue>Cei4QxjWo9Kg8D3pKgxw=</ds:KeyValue>
      </ds:KeyInfo>
    </o-ex:asset>
    <o-ex:permission>
      <o-dd:display>
        <o-ex:constraint>
          <o-dd:count>1</o-dd:count>
        </o-ex:constraint>
      </o-dd:display>
    </o-ex:permission>
  </o-ex:agreement>
</o-ex:rights>
```

Note that the CEK inside the <KeyValue> element is base64 encoded.

8.3.2.2. WBXML Encoding

Token stream (numbers in hexadecimal)	Description
03	WBXML version number – WBXML version 1.3
0E	Public identifier (-//OMA//DTD DRMREL 1.0//EN)
6A	Charset = UTF-8
00	String table length = 00 (empty string table)
C5	<o-ex:rights>
05	xmlns:o-ex=
85	"http://odrl.net/1.1/ODRL-EX" (attribute value)
06	xmlns:o-dd=
86	"http://odrl.net/1.1/ODRL-DD" (attribute value)

07	xmlns:ds=
87	http://www.w3.org/2000/09/xmldsig#/" (attribute value)
46	<o-ex:context>
47	<o-dd:version>
03	STR_I (inline string follows with a terminator)
"1.0", 00	"**1.0**" + string terminator
01	</o-dd:version>
01	</o-ex:context>
49	<o-ex:agreement>
4A	<o-ex:asset>
46	<o-ex:context>
47	<o-dd:uid>
03	STR_I (inline string follows with a terminator)
"cid:4567829547@foo.com", 00	"**cid:4567829547@foo.com**" + string terminator
01	</o-dd:uid>
01	</o-ex:context>
4B	<ds:KeyInfo>
4C	<ds:KeyValue>
C3	Opaque token
10	Length of opaque data (16 bytes = 0x10)
IeC8XrhM9lwfp4Hn	128 bit content encryption key (in binary form, i.e., no base64 encoding)
01	</ds:KeyValue>
01	</ds:KeyInfo>
01	</o-ex:asset>
4D	<o-ex:permission>
4F	<o-dd:display>
52	<o-ex:constraint>
53	<o-dd:count>
"1", 00	"**1**" + string terminator
01	</o-dd:count>
01	</o-ex:constraint>
01	</o-dd:display>
01	</o-ex:permission>
01	</o-ex:agreement>
01	</o-ex:rights>

This example can be encoded in a total of 85 bytes compared to 544 bytes in the XML version. This is a saving of 84%.

Note that the CEK is in binary format (and not base64 encoded) for WBXML encoded rights objects.