



May 2004

Digital Rights Management

Sony Ericsson



Sony Ericsson

Preface

Purpose of this document

This document describes how to protect content using the Digital Rights Management method for Sony Ericsson mobile phones.

The document is intended for content providers and content publishers who want guidelines to protect and publish their premium content.

People who can benefit from this document are:

- Software developers
- Operators and service providers
- Content providers
- Content publishers

More information, useful for product, service, and application software developers, is published on the [Sony Ericsson Developer World Web site](#) which contains up-to-date information about technologies, products, and tools.

These Developers Guidelines are published by:

Sony Ericsson Mobile Communications AB,
SE-221 88 Lund, Sweden

Phone: +46 46 19 40 00
Fax: +46 46 19 41 00
www.SonyEricsson.com

© Sony Ericsson Mobile Communications AB,
2004. All rights reserved. You are hereby granted
a license to download and/or print a copy of this
document.
Any rights not expressly granted herein are
reserved.

Third edition (May 2004)

This document is published by Sony Ericsson Mobile Communications AB, without any warranty*. Improvements and changes to this text necessitated by typographical errors, inaccuracies of current information or improvements to programs and/or equipment, may be made by Sony Ericsson Mobile Communications AB at any time and without notice. Such changes will, however, be incorporated into new editions of this document. Printed versions are to be regarded as temporary reference copies only.

*All implied warranties, including without limitation the implied warranties of merchantability or fitness for a particular purpose, are excluded. In no event shall Sony Ericsson or its licensors be liable for incidental or consequential damages of any nature, including but not limited to lost profits or commercial loss, arising out of the use of the information in this document.

Document history

Change history		
2003-11-01		First edition
2004-01-01		Minor updates.
2004-05-04		Command line extensions, Rights updates

Online Developer Resources

On [Sony Ericsson Developer World](#), developers will find documentation and tools such as phone White Papers, Developers Guidelines, SDKs and APIs etc. The developer Web site also contains discussion forums monitored by our Sony Ericsson Developer Support team, a searchable Knowledge Base of support queries and solutions, Tips & Tricks, example code, and so on. To stay up to date on development issues, register and subscribe to the monthly Sony Ericsson Developer Newsletter.

Sony Ericsson Developer Support

Sony Ericsson offers developers professional technical support services. The service can be purchased from the developer web portal, as part of the Sony Ericsson Core and Core+ membership package, or as individual support incidents. There are two levels of support included in the memberships:

The **Basic E-mail Developer Support** is an annual support service included in the Core membership that provides developers with all the basics to successfully develop world class applications for Sony Ericsson products. With this support contract, developers get access to Sony Ericsson developer support engineers via e-mail with same-day response, five technical support incidents as well as the ability to purchase more.

The **Priority E-mail Developer Support** is an annual support service included in the Core+ membership that equips professional developers with everything they need to successfully develop world-class applications for Sony Ericsson products. With this support contract, developers get priority access to Sony Ericsson developer support engineers via e-mail with fast response times and up to 50 technical support incidents.

Contents

Purpose of this document	2
Document history	3
Online Developer Resources	3
Sony Ericsson Developer Support	3
Introduction	6
DRM overview	8
Business cases for content commerce	9
Preview.....	9
Super distribution.....	9
Subscription	9
How OMA DRM works	9
Forward-lock.....	9
Combined delivery	10
Separate delivery	10
Download descriptors	11
Sony Ericsson DRM Packager	11
Delivery of rights and content	12
Different ways of packaging.....	12
Downloading servers and publishing servers	12
Billing systems	13
DRM technical information	13
Download descriptors	13
Rights & Constraints	14
Forward-lock	15
Combined delivery	18
Separate delivery	20
Sony Ericsson proprietary DRM method	20
Server configuration	21
Forward-lock messages on the web server.....	21
Static web server	21
Dynamic Web server	21
MMS message	22
Sony Ericsson DRM Packager	23
Installation	23
Uninstalling	23
Add/Remove Programs in the Windows Control Panel	23
Start menu.....	23
Graphical Stand-alone version	24
Configuration.....	24
Device profile	25
Main window	26
File menu.....	28
Edit menu	28
Options menu.....	29
Help menu.....	29
Configuration.....	30
Download Descriptors	31
Rights and constraints	32
Rights template file	33

Packaging	33
Command line version	35
Rights generation options	35
Common Command-line options.....	36
Command-line options for all DRM methods	37
Command-line options for forward-lock only	37
Command-line options for forward-lock and combined delivery only.....	37
Command-line options for combined and separate delivery only	38
Command-line options for separate delivery only	38
Device profile	39
MIME types	41

Introduction

This document is intended to give the reader a basic understanding of how DRM is designed and how it is used to protect media. This includes the Sony Ericsson DRM Packager application and associated technology.

It is written for developers that are going to protect media objects residing on their Web server using the OMA DRM standard. It assumes that the reader is familiar with general Web technologies such as HTTP, MIME, PERL, CGI, MMS and Email formats.

More information, useful for product, service and application developers, is published at [DeveloperWorld], which contains up-to-date information about technologies, products and tools.

For more in-depth information on the OMA DRM standard please refer to the standards documents in the "References"-section below.

References

- [DLARCH] Download Architecture Version 1.0, Version 10-June-2002,
OMA-Download-ARCH-v1_0-20020610-p
- [DRMCF] DRM Content Format Version 1.0, Version 13-September-2002,
OMA-Download-DRMCF-v1_0-20020913-a
- [DRM] Digital Rights Management Version 1.0, Version 05-September-2002,
OMA-Download-DRM-v1_0-20020905-a
- [DRMREL] Rights Expression Language Version 1.0, Version 13-September-2002
OMA-Download-DRMREL-v1_0-20020913-a
- [DLOTA] Generic Content Download Over The Air Specification Version 1.0,
Version 12-September-2002
OMA-Download-OTA-v1_0-20020912-a
- [RFC2046] <http://www.ietf.org/rfc/rfc2046.txt>
- [RFC822] <http://www.ietf.org/rfc/rfc822.txt>
- [RFC2396] <http://www.ietf.org/rfc/rfc2396.txt>
- [WSP] "Wireless Session Protocol (WSP)",
<http://www.openmobilealliance.org/documents.html>
- [MMS] "MMS Encapsulation"
- [Conf] "MMS Conformance Document Version: 2.0.0"
- [Developer World] Sony Ericsson Developer World (<http://www.sonyericsson.com/developer>)

OMA documents are downloadable from the OMA website <http://www.openmobilealliance.com>.

Terminology

Input file	Any media file that the user can select to a file list in the Sony Ericsson DRM Packager.
Output file	A file created by the DRM Packager, formatted according to the OMA DRM forward-lock method or Sony Ericsson proprietary forward-lock method, protecting an input file.
DRM agent	A user agent in the device that enforces the rights and controls the consumption of DRM content on the device.
Rights	Permissions and constraints defining under which circumstances access is granted to DRM content.
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.
Combined delivery	Delivery of the rights object and content together in a single message.
Separate delivery	Delivery of the rights object and content via separate transports.
Download descriptor	Metadata about a media object and instructions to the download agent for how to download it.
DRM Packger	The Sony Ericsson DRM Packager. A PC SW based program that packages media objects into OMA DRM format.

DRM overview

What is DRM?

DRM is an abbreviation of Digital Rights Management, a technology that enables secure distribution, promotion and sales of digital media. The purpose of DRM is to make it possible to protect digital content by the means of limited usage and DRM is one of the success factors for 3G, in which operators and content providers need a method to sell content that cannot be freely distributed between devices. The type of DRM described in this document is that defined by the Open Mobile Alliance (OMA) standards listed in the "References" section.

The scope of OMA 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.

Any image, ringtone, or theme in the phone can be sent to another phone by selecting the Send option on the object. A user can "forward" an image received in an MMS message or downloaded from the Web. This is an easy way for users to share popular ringtones and images using transfer methods such as infrared, Bluetooth, email, MMS, and SMS.

What is the problem today?

The right to access content today usually depends on where it is located. The content can, for example, be stored on the Internet, in another mobile phone or in other devices. The content itself is unprotected. Most pictures and ringtones are available on Web sites in their raw unprotected format, ready for immediate use in the device. It is easy for anyone to download and save content in the device for free. This means that it remains difficult to create a large market where money is exchanged and revenues are generated.

A Web site providing pictures and ringtones for download may grant access only to subscribers. When the content is in the device, the user often has free access and can copy it to other devices. Once a picture or ring signal is in a device, the content provider has no control over the content. High-value content is kept away from the market as long as its usage cannot be controlled, which is clearly a disadvantage for all involved.

What can DRM solve?

Usage rights can be defined for all types of content. Different rights can be applied to the same content for different users. Here are some examples of rights that can be controlled with DRM:

- the number of times it can be used, for example, a game can be played five times
- the period it can be used, for example, until 31st December 2004
- the length of time it can be used, for example, two weeks or one day

This control enables content providers to create more sophisticated subscription and pay-per-use models, than can be done today. This is clearly a benefit for content creators (the artist that records the music or the graphic designer that creates a new wallpaper), content providers (who make the content available to the user), network operators, and everyone else involved. It will also stimulate the mobile content based services. Sony Ericsson DRM solutions have been developed with these success factors in mind.

Business cases for content commerce

Three typical business cases are described below.

Preview

One example of a business case is to preview content. A content provider wants to offer the opportunity to try a game written in Java for free, hoping that the user will find it attractive and buy it afterwards. With a DRM solution, the only thing the content provider has to do is to package the game in a DRM package with the rights to execute it only once. The user downloads the package, and the game is automatically installed, the same way any Java game would be. The device keeps track of the number of times the game is executed, and if the user tries to start it a second time, it will inform the user that this was a preview only. The device will assist the user with how to obtain rights to use it for a longer period of time.

Super distribution

Another example is a way to spread content to users who would not normally have found it. One user has bought a great wallpaper. His or her friend would like a copy of it. As it was downloaded long ago, the user cannot remember where it was found. Being encrypted, though, it will be OK to forward it by beaming via Bluetooth or infrared. This is called super distribution. The friend will receive the file in his/her device and will also be assisted automatically with how to obtain rights to use it. With OMA DRM, it is possible for the content provider to enclose different rights with the same content, since the content and the rights are delivered in separate files.

Subscription

It will be possible for users to subscribe to certain services, such as streaming videos from sporting events. The user might, for example, choose to subscribe for one week at time. Each week the operator sends out the rights to use the service for another week, and the cost is automatically added to the user's account. The user can terminate the subscription whenever he or she wants.

How OMA DRM works

The importance of an open standard

Sony Ericsson is actively focusing on technology standardization for the DRM concept. Sony Ericsson is fully committed to open standard solutions in the mobile environment and is a principal driver of many open standard initiatives. This will ensure the interoperability of mobile terminals in the DRM area and also result in a strong, competitive DRM standard at the earliest possible stage.

An open standard will limit a fragmented market where manufacturers support different DRM solutions. Non-standard solutions, where content providers and operators have to develop parallel infrastructures that support different DRM solutions, will slow down the market development. It will also create interoperability issues for content providers. The OMA DRM standard will limit the need to support multiple DRM solutions.

The current OMA DRM standard offers three means of content protection:

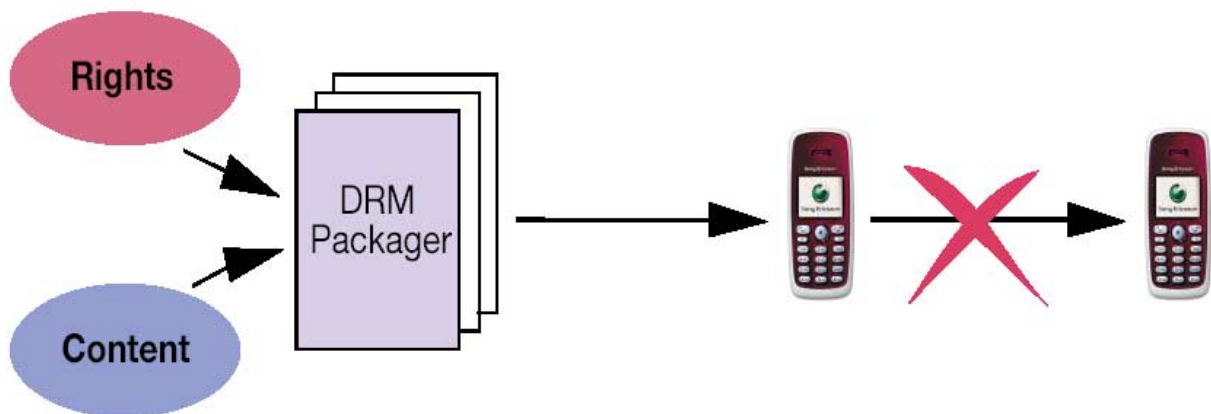
Forward-lock

Forward-lock is a special case of combined delivery (see the next section).

Combined delivery

Rights and content are packaged together into one DRM package and delivered to the device. In the simplest case, no special rights are defined. The content is just put into a DRM package, thus protected from being copied out from the device by the user. This special case is called "forward-lock". It is useful for all types of content that the provider wants to charge for.

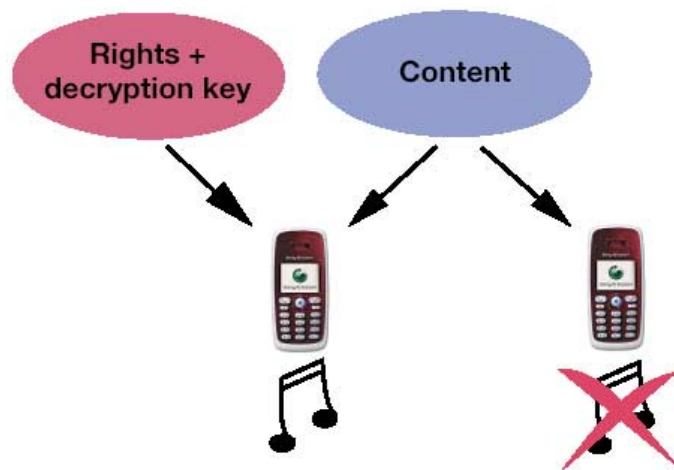
Forward-lock is not, the way it is specified by the OMA DRM standard, a secure way of protecting content on the Internet in general. An object inside a message is not encrypted. Anyone with a PC and knowledge of the message format (available on the Web for free) can open the message, retrieve the object, and use it without restrictions. When the message is inside the phone, this is difficult to do. When downloaded to a PC, this is easy to do.



Combined delivery

Separate delivery

Rights are defined and put into a file of their own. The content is encrypted and made available for users to download to their devices. The decryption key is put into the rights file. Since the content is encrypted, users cannot access it before the rights are also in the device. In this case, the content can be freely distributed on the network, only users with the rights file can access the content. Content providers can deliver the rights to the user using push technology.



Separate delivery**Download descriptors**

The download descriptor is a collection of attributes, used to describe a media object at a URI or URL (as defined in [RFC2396]). The defined attributes are specified to allow the download agent to identify, retrieve, and install media objects.

A predefined set of attributes is specified to allow the download agent software to identify, retrieve, and install media objects. All attributes appearing in the download descriptor are made available to the content handler of the media type that the download descriptor references.

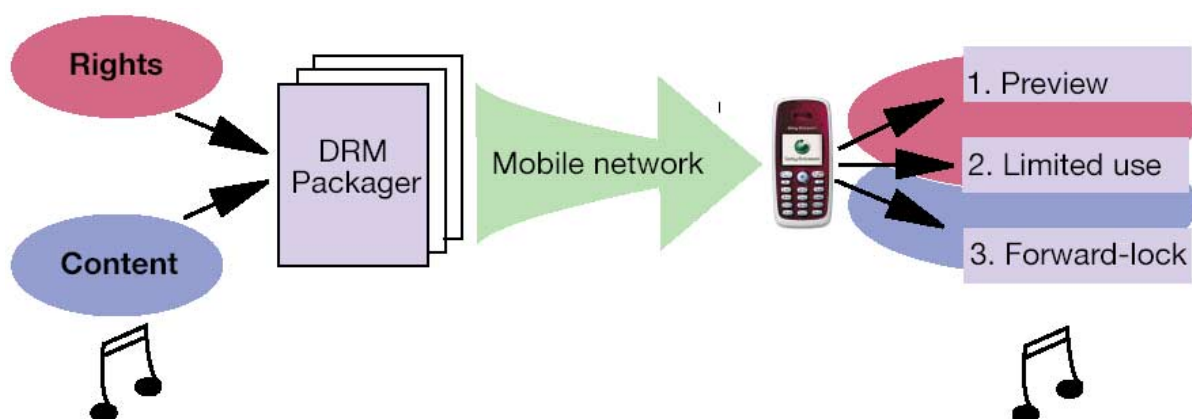
The descriptor allows the device to verify that the desired media object is suitable for the device before being loaded. It also allows media object-specific attributes to be supplied to the relevant content handler. The client device use the MIME media type declared by the transport or packaging mechanism to identify a download descriptor object.

Other standardization organizations work with various aspects of DRM but the OMA standard is specifically targeted at mobile devices. The specifications are available at <http://www.openmobilealliance.org/documents.html>.

Sony Ericsson DRM Packager

The Sony Ericsson DRM Packager, hereafter referred to as the "DRM Packager" is used to create the DRM package that is delivered to the device, including content and associated rights. The DRM package, containing rights and content, gets from the content provider to the device in the same way as any other content on the Web. In the device, the DRM Package is unpacked, and the content is made available to the user according to the rights. If the rights permit the user to play a ringtone ten times, the device will keep track of the number of times the ringtone is played, and notify the user when the ringtone has been used for the tenth time.

Content protection according to the OMA DRM standard gets special properties. Unless the content is encrypted, the user cannot copy DRM content to other devices since the "Send to" option is disabled for pictures, ringtones, etc. that are OMA DRM protected. Content providers may choose to protect some content, but leave some content unprotected.



Sony Ericsson DRM Packager

Delivery of rights and content

Different ways of packaging

Rights and content can be packaged together and delivered to the device as one DRM package. As an alternative, content can be delivered to the device first, followed by the rights later being pushed to the device, for example, via SMS. The kind of service and business model adopted by the content provider determines how the content and rights should be packaged and delivered to the device.

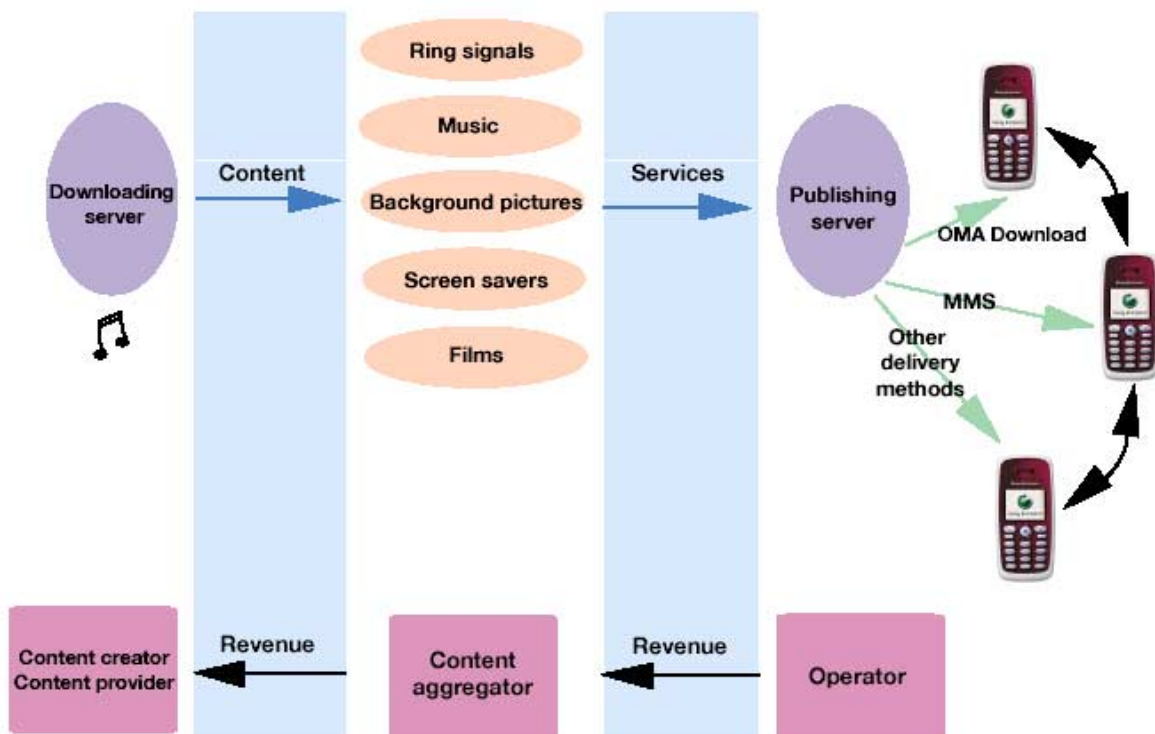
Downloading servers and publishing servers

When using a mobile phone, the users do not have to be aware of the network architecture. During a content downloading session, typically many physical servers are involved. Sometimes transactions may take place between different companies' servers.

The actual content may be put on one server, the downloading server. The content can be reached, e.g., through references from one or many other servers, the publishing servers. The content creator puts his or her content on the downloading server through an interface to the content provider.

The user navigates to the publishing server and selects the content, or rather a link to or description of the content. The content is then downloaded from the actual downloading server.

When content is downloaded to the device, operators generate revenues from the user via e.g. their billing system. Operators might in their turn be billed for rights by the content aggregator, content provider or directly by the content creator.



The flow of revenues and content. The content is viewed and selected from a publishing server and downloaded to the mobile phone from a downloading server. The revenue is in this case collected from the user by the operator and transferred to the content creator via the content aggregator.

Billing systems

Billing is actually not a part of DRM solutions even though it is one of the success factors. OMA has also, however, standardized a download mechanism that can be used to enable different billing models. Users need not necessarily have to pay for DRM protected content. A library can, for example, use DRM for e-Books. An operator can use DRM to provide free previews (play once) of ringtones and screen savers.

DRM technical information

Download descriptors

Download descriptors are not dependent on the DRM method used, except for the need to define the proper MIME types that the destination device needs to support in order to handle protected content of the DRM method that protects the content. Download descriptor files have a file extension of ".dd". For more information on download descriptor files, please refer to [DLOTA].

It is recommended to use the Sony Ericsson DRM Packager to automatically generate download descriptor files tailored to the protected media content they describe (see "Download descriptors" on page 31).

The following is an example of a download descriptor file:

Download descriptor file	
<code><media xmlns="http://www.openmobilealliance.org/xmlns/dd"></code>	Standard tag containing rest of DD.
<code><objectURI>http://download.example.com/image.gif</objectURI></code>	URI used to access media object.
<code><size>29753</size></code>	Size of media object is 29753 bytes.
<code><type>image/gif</type> <type>application/vnd.oma.drm.rights+wbxml</type> <type>application/vnd.oma.drm.content</type></code>	Media object is a gif image. The client device must support additional MIME types needed by DRM method (in this case, separate delivery).
<code><name>Beach Image</name></code>	A user readable name of the media object that identifies the object to the user.
<code><vendor>The example corporation</vendor></code>	The organisation that provides the media object
<code><description>This is an image of a beach.</description></code>	A short textual description of the media object

<code><installNotifyURI> http://download.example.com/notifyscript?file=image.gif </installNotifyURI></code>	URI to which a installation status report is to be sent on success as well as on failure.
<code><nextURL>ff</nextURL></code>	URL to which the client should navigate if the user continues browsing after download
<code><infoURL>dd</infoURL></code>	URL that further describes media object
<code><iconURI>cc</iconURI></code>	The URI of an icon
<code><installParam>hh</installParam></code>	Installation parameter for media object
<code></media></code>	<i>End of standard tag</i>

Rights & constraints

The usage of a DRM file is controlled by one or more rights objects when using separate or combined delivery. Note that these rights are not compatible with all MIME types. All of these rights are specifically compatible with only certain MIME types. There are the four usage rights:

Constraint	
Display	The display right has the semantics of rendering the DRM content onto a visual device, for example, image/gif or image/jpeg.
Play	The play right has the semantics of rendering the DRM content into audio/video form, for example, audio/midi, video/quicktime.
Execute	The execute right has the semantics of executing, i.e. invoking, DRM content, e.g. Java games or other applications.
Print	The print right has the semantics of printing, i.e. creating a hard-copy of, the DRM content, for example, image/jpeg.

A usage right may be limited by one or more of the following constraints:

Constraint	
Count	The usage right may only be exercised a limited number of times.
Interval	The usage right may only be exercised for a specified time interval, starting the first time the right is exercised.
Datetime	The usage right may only be exercised within a specified period of time, with start and end time specified.

Forward-lock

Output Files

The content file to protect is wrapped in a “wrapper file” of MIME type "application/vnd.oma.drm.message".

Forward-lock content files have a file extension of ".dm"

Message format

The forward-lock message has the same syntax as a MIME "multipart", which is defined in detail in [RFC2046] section. Forward-lock is a special case of combined delivery where no rights have been defined. Please refer to the "combined delivery" section below to get additional information on what the message format looks like when rights have been defined for a media object.

The following is an example of HTTP transfer of a forward-locked media object from a Web server to a client device:

HTTP-Response	
<i>HTTP/1.1 200 OK</i> <i>Server: SomeServer</i> <i>Content-Length: 3218</i>	The HTTP response code followed by header fields.
<i>Content-Type: application/vnd.oma.drm.message;</i> <i>boundary=XYZABC</i>	Content-Type indicates Forward-lock object
	One empty line, ended by CRLF.
--XYZABC	The boundary delimiter, ended by CRLF.
<i>Content-Type: image/gif</i> <i>Content-Transfer-Encoding: binary</i>	Headers belonging to forward-lock object.
	One empty line, ended by CRLF.
XX XX XX XXxx	The binary encoded object.
--XYZABC--	Final boundary delimiter followed by empty line, i.e. starts and ends with CRLF.

The forward-lock media type requires one parameter, the boundary parameter, which is part of the boundary delimiter line. The boundary delimiter line is then defined as two hyphen characters ("-") followed by the value of the boundary parameter, optional white-space, and a terminating CRLF. The boundary delimiter line always occurs at the beginning of a line, that is, after a CRLF. The CRLF is considered to be part of the boundary delimiter line. After the terminating CRLF there is either the header fields of the next part, or another CRLF, in which case the next part has no header fields.

The boundary delimiter must be no longer than 70 characters, not counting the two leading hyphens. As a delimiter, the boundary delimiter line must not appear inside the encapsulated object.

The last boundary delimiter is identical to the other delimiters except that it has two more hyphens at the end.

Mandatory encodings according to [OMADRM] are the default 7-bit, 8-bit and binary, and those encodings should be used to achieve best possible interoperability. However, in environments with systems that support only 7-bit ASCII characters, such as some old email servers, it is necessary to base64 encode the object before it is put inside the message.

But base64 encoding increases the size of the object by 33%. In a Web and WAP environment, with a modern Web server and a WAP gateway as the intermediate nodes between the client and the server, base64 encoding is unnecessary. On the other hand, base64 encoding object guarantees that the boundary delimiter line will not appear inside, hyphen characters are not part of the base64 encoding.

Note: The forward-lock message is actually just a special usage of the general DRM message, defined by the OMA DRM standard. The message can be used for more advanced DRM features, not covered in this document, in addition to forward-lock.

The Content-type field and the boundary parameter

When the forward-lock message is transported in an HTTP response, the media type name and the boundary parameter are inside the Content-type field. The grammar of the HTTP Content-type field is defined in RFC2616:

```
Content-Type = "Content-Type" ":" media-type
media-type = type "/" subtype *( ";" parameter )
type = token
subtype = token
parameter = attribute "=" value
attribute = token
value = token | quoted-string
```

The forward-lock media type has one required parameter, the boundary parameter.

WAP implementation

Note: Parameters are rarely used in media types. An exception is the "multipart" media types (e.g. multipart/related), in which parameters are used in the same way as in the forward-lock message – the forward-lock message has actually the same syntax as a multipart media type. Existing WAP gateways can encode multipart media types, because the WAP protocol has a special encoding for such messages. The forward-lock message is a new message type defined by OMA. Unknown media types are encoded in plain text. How to encode is defined in the WAP Wireless Session Protocol [WSP] specification, the uncertainty comes from the fact that this function in WAP gateways has, until now, never been used.

The boundary parameter is formally defined by the following BNF [RFC2046]:

```
boundary := 0*69<bchars> bcharsnospace
bchars := bcharsnospace / " "
bcharsnospace := DIGIT / ALPHA / "'" / "(" / ")" /
"+ " / "_" / "," / "-" / "." /
"/" / ":" / "=" / "?"
```


Due to the grammar of the Content-type field it is sometimes necessary to enclose the boundary parameter value in quotes. A typical Content-type header field might look like this:

```
Content-Type: application/vnd.oma.drm.message; boundary=gc0p4Jq0M2Yt08j34c0p
```

But the following is not valid (because of the colon):

```
Content-Type: application/vnd.oma.drm.message; boundary=gc0pJq0M:08jU534c0p
```

It must instead be represented as:

```
Content-Type: application/vnd.oma.drm.message; boundary="gc0pJq0M:08jU534c0p"
```

WAP Implementation

Note: Avoid using boundaries with large numbers such as "222222222222222222". Some gateways, when converting it into WAP binary protocols, interpret the boundary as a large integer instead of a string; which will result in an error in the client.

MMS format

MMS content can be forward-locked by putting individual media objects within forward-lock envelopes. A complete MMS message cannot be forward-locked with an OMA DRM envelope; instead, each individual media object needs to be forward-locked. Multipart objects should not be included in this OMA DRM envelope. A device that supports only forward-lock must discard DRM messages that contain a rights object, see the OMA DRM specification.

The SMIL presentation should reference the DRM content through a Content-Location or Content-ID applied to the DRM message (please note that this is a header field different from an optional Content-ID inside the DRM message – but may contain the same value). The MMS conformance paper [Conf] specifies that the MIME type for MMS message content should be multipart/related or multipart/mixed.

Just as in the case with OMA forward-lock over WSP some information is binary encoded and some information is not binary encoded.

One forward-locked media object

This example shows the textual representation of an MMS message with a forward-locked media object (a jpg image).

```
Content-Type:multipart/related;boundary=ID_1234567
```

```
--ID_1234567
```

```
Content-Type:application/smil
```

```
<smil>
```

```
:          
```

```
</smil>
```

```
--ID_1234567
```

```
Content-Type:application/vnd.oma.drm.message;boundary=PUTJTBYRBYTYBV
```

```
Content-Location:drmimage.dm
```

```
--PUTJTBYRBYTYBV
```

```
Content-Type:image/jpeg
```

Content-Transfer-Encoding:binary

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
--PUTJTBYRBYTYBV--
--ID_1234567--
```

The textual representation follows RFC822 very closely. When the message is sent to the client, the MMS protocol specifies a binary format [MMS]. The MMS format uses WSP rules to encode headers and multipart objects. Because OMA DRM does not specify a binary encoding format of the application/vnd.oma.drm.message MIME type, these objects must be transferred in their textual form.

Header and multipart boundary delimiters (--ID_1234567) will be translated to their binary encoding. Multipart boundary are translated to multipart entries specifying number of objects. Since there is no binary encoding of the OMA DRM envelope format, header and boundary delimiters (--PUTJTBYRBYTYBV) for the forward-locked media object will not be translated to any binary encoding. This means that the client has to handle textual boundary delimiters and cannot rely on WSP encoding.

Note: It is impractical to support forward-lock for every media type. In most cases only a few are used in MMS messages, and are relevant to forward-lock. There are media types such as vCard that may be of no interest to forward-lock.

All image formats (e.g. GIF, JPEG, WBMP) and sound formats supported by the MMS client can be "forward locked" by putting the object into a OMA forward-lock message.

Combined delivery

Output files

The content file to protect is wrapped in a "wrapper file" of MIME type "application/vnd.oma.drm.message".

Combined delivery protected content files have a file extension of ".dm".

The wrapper file also contains a set of rights and constraints, in a rights object of MIME type "application/vnd.oma.drm.rights+xml". This rights object describes how the content file may be used as described in [DRMREL].

Message format

Since forward-lock is a special case of combined delivery, the format of a combined delivery message is very similar to forward-lock. They both have the same syntax as a MIME "multipart", which is defined in detail in [RFC2046]. The difference is that a combined delivery message includes rights along with the media object. The rights are thus a part of the multipart body and have the MIME-type "application/vnd.oma.drm.rights+xml".

The following is an example of HTTP transfer of a combined delivery protected media object from a Web server to a client device:

HTTP-response	
HTTP/1.1 200 OK Server: SomeServer Content-Length: 3218	The HTTP response code followed by header fields.
Content-Type: application/vnd.oma.drm.message; boundary=XYZABC	Content-Type indicates combined delivery
	One empty line, ended by CRLF.
--XYZABC	The boundary delimiter, ended by CRLF.
Content-Type: application/vnd.oma.drm.rights+xml Content-Transfer-Encoding: binary	Headers indicating rights connected to media object below.
	One empty line, ended by CRLF.
<pre><o-ex:rights xmlns:o-ex="http://odrl.net/1.1/ODRL-EX" xmlns: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:unique-id-2187617881@my.domain</o-dd:uid> </o-ex:context> </o-ex:asset> <o-ex:permission> <o-dd:display/> </o-ex:permission> </o-ex:agreement> </o-ex:rights></pre>	<p>"uid" and "Content-ID" refer to the content id shared by the media object and the rights object, connecting them together.</p> <p>It is a global id valid in any part of the world, consisting of a locally generated id and a domain name.</p> <p>One right has been defined for the gif image appearing below, the display right. No constraints on this right have been defined.</p> <p>Please refer to [DRMREL] for a detailed explanation of the rights language used here (Rights Expression Language).</p>
	One empty line, ended by CRLF.
--XYZABC	The boundary delimiter, ended by CRLF.
Content-Type: image/gif Content-ID: <unique-id-2187617881@my.domain> Content-Transfer-Encoding: binary	Headers define media object type and content id, connecting media to rights defined above.
	One empty line, ended by CRLF.
XX XX XX XX	Binary encoded media object.
--XYZABC--	Final boundary delimiter followed by empty line, i.e. starts and ends with CRLF.

Separate delivery

Output files

The content file to protect is wrapped in a “wrapper file” of MIME type "application/vnd.oma.drm.content".

Separate delivery protected content files have a file extension of ".dcf".

The content file within the ".dcf"-file has been encrypted, and the file can therefore be sent (super distributed) from the phone without allowing a non-licensed client device access to its contents. Unlike a ".dm"-file, the rights object is not included in the wrapper ".dcf"-file. Instead it is pushed out to the phone, using for instance, SMS. The rights are contained in

- ".dr"-files (MIME-type "application/vnd.oma.drm.rights+xml")
- or
- ".drc"-files (MIME-type "application/vnd.oma.drm.rights+wbxml")

These files contain rights of usage granted to the user that downloads the content ".dcf"-file. ".dr"-files are in XML text format (same as in combined delivery) and ".drc"-files are in WBXML binary format.

Message format

The encryption features and special file format of the separate delivery method makes this method more complicated and needs more detailed documentation to be fully explained.

Please refer to [DRMCF] for more information on the format of this type of protected content.

Please refer to [DRMREL] for detailed information on rights formats.

It is recommended that one uses the Sony Ericsson DRM Packager to protect content using this DRM method. Please refer to the "Sony Ericsson DRM Packager" for more information on how to do this.

Sony Ericsson proprietary DRM method

Output files

Sony Ericsson have developed a proprietary DRM method not included in the OMA DRM standard but supported by the Sony Ericsson DRM Packager. This method consists of a proprietary Sony Ericsson DRM header added to the content. It does not feature more sophisticated DRM management such as the encryption and rights features as used with combined and separate delivery.

Sony Ericsson protected content files have a file extension of ".copy".

The MIME type is "application/vnd.sem.mms.protected".

Message format

This information is proprietary of Sony Ericsson.

Server configuration

Forward-lock messages on the web server

A forward-lock message on a Web server can be either created dynamically at each request or served as a plain text file. As long as the message exactly follows the defined syntax, it does not matter how it is stored and how it was created on the server.

The following subsections explain two ways of creating forward-lock messages on the Web server. The procedure for combined delivery protected files is the same.

Static web server

Here is how to create a forward-lock file and how to configure the Web server to serve it up as a forward-lock message. The procedure for combined delivery protected files is the same.

Step 1:

Create a forward-lock message with "mime_content_boundary" as the boundary and save it to the file system with ".dm" as the file extension.

Note: The message can be created using the Sony Ericsson DRM Packager (see section 6).

The following example forward-lock message encapsulates a GIF image.

```
--boundary1
Content-Type: image/gif
Content-Transfer-Encoding: binary
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
--boundary1--
```

Step 2:

On the Web server, map the ".dm" file extension to the forward-lock media type and the "boundary1" boundary parameter. If you are using Apache, the following entry should be in the "mime.types" file:

application/vnd.oma.drm.message;boundary=boundary1 dm

The boundary parameter does not have to be "boundary1". It can be any legal parameter, as long as it is the same in the parameter of the media type name as it is in the forward-lock message.

Note: Some servers do not accept mapping between a media type with a boundary and file extensions. This has, however, been successfully tested on the Apache and Microsoft web servers.

Dynamic Web server

A dynamic Web server is implemented as a program (written in any Web server language such as PERL, ASP, etc) that, for each request, creates a forward-lock message.

This is more complicated for combined delivery protected media as rights need to be defined. We recommend that one uses the Sony Ericsson DRM Packager to create preprotected files and publishes them using the procedure described above.

MMS message

Creating a forward-locked media object within an MMS message is very similar to how static Web media objects are created.

Step 1:

Create a forward-lock message with "boundary1" as the boundary, and save it to the file system with ".dm" as the file extension.

```
--boundary1
Content-Type: image/gif
Content-Transfer-Encoding: binary
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
--boundary1--
```

Step 2:

Add the forward-locked object to the MMS message. Set the correct MIME type and assign the boundary 'boundary1' "application/vnd.oma.drm.message;boundary=boundary1". Add Content-Location or Content-ID header with the appropriate value, URI or filename, if referenced in SMIL presentation.

Sony Ericsson DRM Packager

The Sony Ericsson DRM Packager, hereafter referred to as the "DRM Packager", is a PC software based tool which enables third party content providers/developers to protect their content. This tool will focus only on the 'packaging' capability of content protection. The DRM Packager does NOT provide any additional functionality, such as components in the network, e.g. downloading servers, publishing servers or OTA download mechanisms.

Installation

The DRM Packager comes in two versions:

- A graphical Stand-alone Windows application
- A command line utility

The platforms supported are Windows NT/2000/XP.

To install the DRM Packager, simply run the setup program (setup.exe). During the installation, a destination directory in the default Windows program files directory ("Program Files") will be suggested ("Sony Ericsson\DRM Packager"). The user is free to select this or any other destination path.

A folder under the start menu will also be suggested ("Sony Ericsson\DRM Packager") where links will be created to the application executable and uninstallation script.

Uninstalling

When you uninstall the DRM Packager all installed files are removed along with registry keys etc. If the user has left files in the application folder or any of its subfolders, those files and folders will remain after uninstallation. Note that main.dpr WILL also be deleted! It is therefore better to create a new profile file if the user wishes to edit it, since the uninstall script will not touch the new file, rather than edit the default file ("main.dpr").

There are two ways of uninstalling DRM Packager from your computer:

Add/Remove programs in the Windows control panel

Start Add/Remove Programs in the Windows Control Panel and select "Sony Ericsson DRM Packager". Then click the uninstallation button to begin the uninstallation process.

Start menu

Go to the "Sony Ericsson\DRM Packager" folder in the "Programs" section of the Start Menu and select "Uninstall DRM Packager".

Graphical stand-alone version

To create protected content, the DRM Packager needs the following items:

- Input file list of content files
- Configuration options

Input file list

The input file list contains the following information:

- File path to the input file.
- MIME type of the input file.
- Embedded name - The name encoded in the headers of the MIME body part. Used only if the "Embed content name"-configuration described below is chosen.

It is possible to load and save input file lists stored under the file extension .lst.

Configuration

Global options

Option	
DRM Method	What method to use to protect the content. Currently four versions are available: forward-lock, Sony Ericsson proprietary, combined delivery and separate delivery.
Output directory	The directory where the protected versions of the content files will be stored.
Place output in DRM subfolder under original content folder	This is a binary yes/no-option. If it is set to yes, then an output file will be stored in a subdirectory, called "DRM", to the directory where its corresponding input file is located. I.e., input files in "D:\content\" will generate output files in "D:\content\DRM\". If it is set to no, then the all output files will be stored in the output directory defined in the previous configuration option.
Generate Download Descriptor (DD) file	Generate a download descriptor file for each content file as defined in [DLOTA].

Options for forward-lock and combined delivery

Option	
Encoding type	The MIME content encoding type used to encode the data. Currently four types are available: binary, 7bit, 8bit and base64.
Embed content name	This is a binary yes/no-option. If it is set to yes then the name of the input file will be encoded into the MIME body part headers of the output file. The embedded name for the input files can be manually overridden by the user.
User Defined Boundary	This is a binary yes/no-option with an additional string parameter. If it is set to yes, then the string parameter entered by the user will be used as the boundary separating the different body parts in the MIME body of the output file. If this option is set to no, then a default boundary will be used.

Device profile

Mobile devices are categorized under different device types defined in a device profile. A device profile provides the DRM Packager with information regarding the capabilities supported by the devices of each device type defined in it. Choosing a particular device type limits the configuration options available to the user, making it impossible to configure the software in such a way as to allow creation of protected content that will be unsupported by the target devices belonging to that device type. See example of a device profile on page 39.

Device profiles are XML-based files and are stored under the file extension .dpr.

A default profile ("**Profile\main.dpr**"), provided with the application, is always loaded upon application start-up.

A device profile contains the following information:

- Profile name
- Profile version.
- Minimum DRM Packager version needed to parse the device profile.
- One or more device type definitions.

Device type

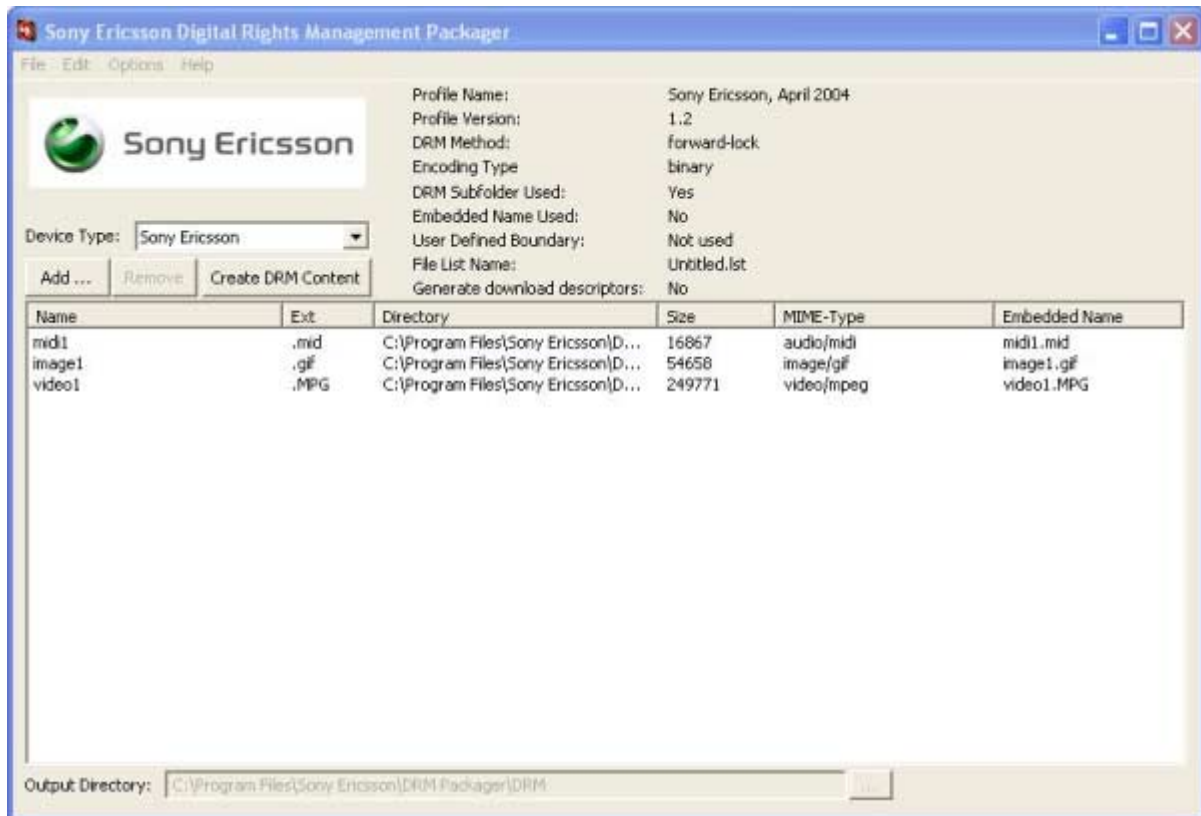
A device type definition contains the following information:

- Device type name
- One or more DRM methods supported by the device type.
- One or more encoding types supported by the device type, e.g. binary, 7bit, 8bit and/or base64.

- One or more MIME types supported by the device type and their corresponding file extensions, used by the DRM Packager not only to determine which types are supported by the device type, but also to automatically map file extensions to MIME types in the input file list.

For each MIME type, one may define none or up to four compatible DRM rights as defined by [DRM-REL]: display, print, play and execute. The combined delivery and separate delivery DRM methods that generate information about a user's rights to the content may only generate rights that are compatible with the MIME type of the content as defined in the device type.

Main window



Text labels

At the top part of the main window there are text labels showing the:

- Current name and version of the device profile used.
- Name of the currently open list file. If no list file is open, the name "untitled.lst" is shown instead.
- Current configuration settings.

The device type drop down list allows the user to select which of the device types defined in the device profile to use. This limits the available configuration options and assures the compatibility of the protected content files with the target devices.

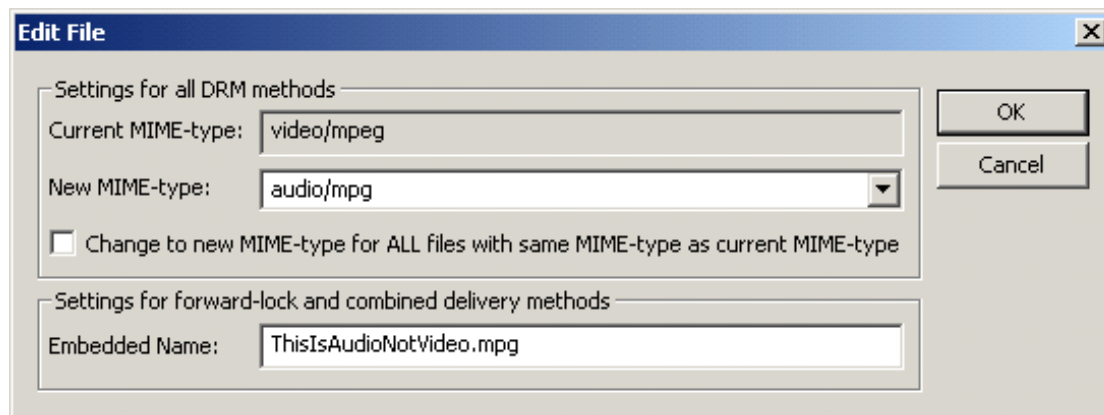
File list

The input file list consists of six columns, which can be sorted alphabetically by clicking on their headers, and contains the following information:

Column header	Description
Name	Input file name without extension.
Ext	Input file name extension.
Directory	Input file directory.
Size	Input file size.
MIME type	Input file MIME-type. This column is automatically filled in by the DRM Packager by using information from the list of supported MIME-types in the definition of the currently chosen device type. If a MIME-type is not supported this field is left blank.
Embedded name	The name encoded in the headers of the MIME body part. Used only if the corresponding configuration option is set.

The user can select files with the mouse by either dragging the mouse with the left mouse button pressed down to create a bounding box as can be done on the Windows desktop or by clicking on a file entry. Holding the CTRL-key down while one operates the mouse allows one to toggle selection status of the files one desires without altering the selection status for the rest of the files in the list.

By double-clicking on a file it is possible to change the embedded name and the MIME-type of the file effectively overriding the system mapped MIME-type and embedded name. By using the checkbox "Change to new MIME-type for ALL files with same MIME-type as current MIME-type" it is possible to change the MIME-type for more than one file at a time. The dialog looks this:



While using the DRM methods Forward-lock and Sony Ericsson proprietary, it is possible to type in an arbitrary MIME-type or select one from a drop down box with MIME-types specified in the profile.

While using the DRM methods combined and separate delivery, it is ONLY possible to select a MIME-type from the drop down box, i.e. the user cannot type in an arbitrary MIME-type. Also, only MIME-types associated with at least one compatible right will be shown in the drop down box. Because the list, while these

DRM methods are active, can only contain one MIME-type, the checkbox described earlier will be automatically checked and non-editable. This will make sure that the MIME-type is changed for all files in the list.

Note: If you change the device type, the MIME type will be reset to show compatibility with the new device type

Buttons

The "Add"-button or "Insert"-key opens a file dialog that allows the user to select input files to add to the list. Files can also be added to the file list by drag-n-drop from the explorer.

The "Remove"-button or "Delete"-key removes selected files from the file list.

The "Create DRM Content"-button starts the DRM Packaging process whereby the content is protected.

Output path edit box

At the bottom of the main window, the output directory can be edited by either entering a directory path in the edit box or by clicking on the "..."-button to open a folder selection dialog. The default output directory is a subfolder called "DRM" to the application folder, e.g. for example "C:\Program Files\Sony Ericsson\DRM Packager\DRM".

File menu

Commands found on the **File** menu:

Command	Function
Add files...	Add files into current file list.
New list	Create a new file list.
Open list...	Open an existing file list.
Save list	Save an edited file list.
Save list as...	Save a file list as another one.
Open device profile...	Open a device profile.
Create DRM content	Start the DRM Packaging process whereby the content is protected.
Close	Save changes and close DRM Packager.

Edit menu

Commands found on the **Edit** menu

Command	Function
Remove selected	Remove selected files from current file list
Remove all	Remove all files from the current file list

Options menu

Commands found on the **Options** menu

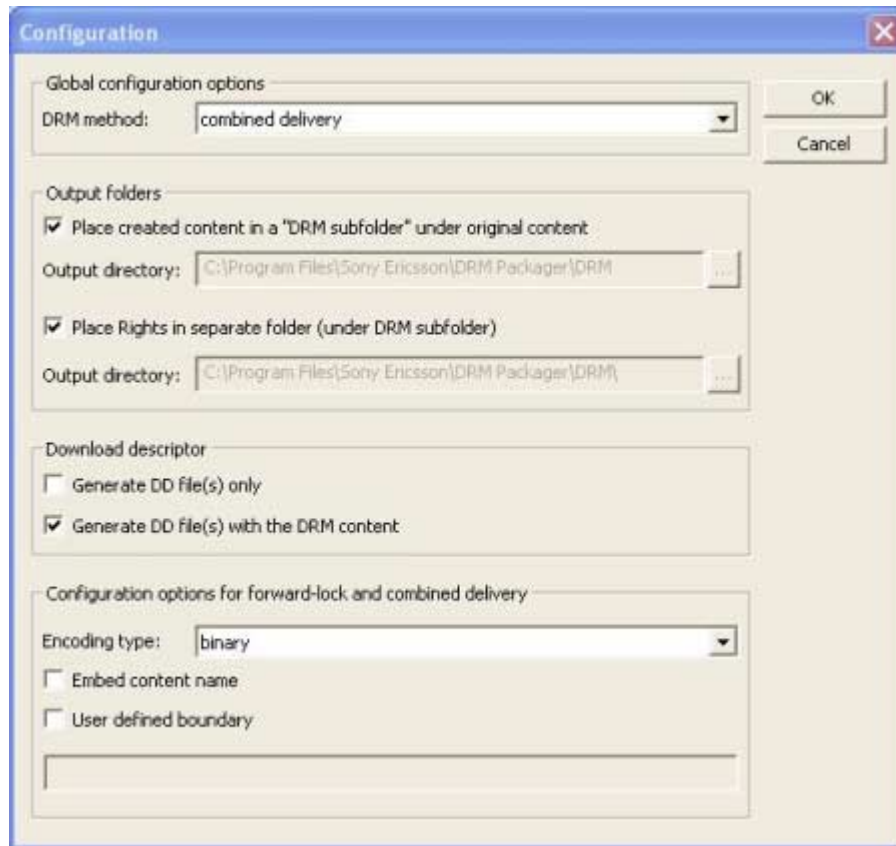
Command	Function
Configuration ...	Open dialog to edit the DRM Packager configuration options.
Rights ...	Open dialog to define rights used when protecting content using combined delivery and separate delivery. This menu option may be disabled should the combined delivery or separate delivery not be the DRM methods currently chosen.
Download descriptor (DD) ...	Open dialog to define information on how to generate download descriptors for the content files. This menu option may be disabled should the generation of download descriptors be disabled in the configuration dialog.

Help menu

Commands found on the **Help** menu

Command	Function
About...	Display an about box containing the application's version number, including copyright information

Configuration



Using the configuration dialog box the user can easily select the desired configuration options as described in the "Overview"-section above.

The combined delivery and separate delivery methods are not compatible with a file list containing multiple MIME types. If one chooses one of these methods while the list contains multiple MIME types, the list will be cleared.

The first time a user chooses to enable the "Generate Download Descriptor (DD) file"-option within a session, the application will display a message box with a reminder that the download descriptors dialog with the necessary options for generation of descriptors are available in the "Options"-menu.

Directory configuration options can be provided instructing the DRM Packager where to store the generated contents. The following options are provided:

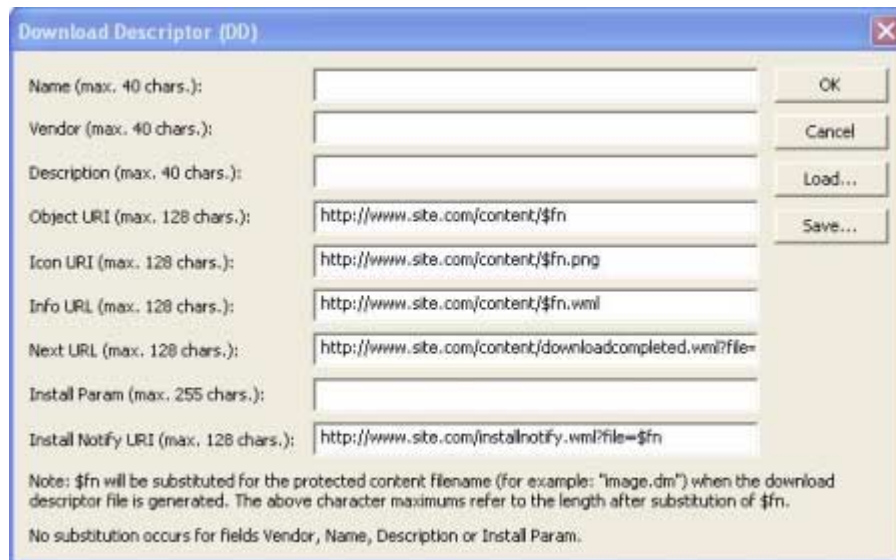
- where to store the DRM generated content (.dm and .dcf) files
- where to store the generated rights (.drc) files
- whether to generate download descriptors (.dd) exclusively (without DRM generated content) or not
- encoding type, user defined boundary and embed content name options

Download descriptors

This dialog defines the way download descriptor files (*.dd) are generated for the content files. Download descriptors are described in detail in [DLOTA]. Fields described in [DLOTA] but not featured in this dialog are automatically generated by the application and need not concern the user.

White space preceding or following the input of the fields is automatically eliminated. The resulting download descriptor XML content will not include tags whose corresponding dialog fields are empty or contain only white space.

To generate unique download descriptors, some fields feature automatic string substitutions. One can put the string "\$fn" in any of the fields Object URI, Icon URI, Info URI, Next URL or Install Notify URI. Upon generation of a download descriptor file for a particular content file, the "\$fn"-string will be substituted for the name including the file extension of the protected content file (for example, myprotectedcontent.dm). The Object URI field in the download descriptor generated from the example dialog below for the content file "myprotectedcontent.dm" would become "http://www.site.com/content/myprotectedcontent.dm".



Object URI is a mandatory field according to [DLOTA]. All other fields are optional. This dialog will not accept an empty Object URI field.

Please note the length limits for the strings as described in the dialog for each field. These size limits originate from [DLOTA] and apply to the data of the download descriptor tags AFTER string substitutions have been performed. Please make sure that these limits are abided by and remember that long file names may cause non-conformance even if the fields of the actual dialog are conformant.

Download descriptor template file

By using the "Load..."- and "Save..."-buttons in the dialog above it is possible to load or save a download descriptor template file. This type of file represents the information contained in the dialog in an XML format. It can be used to quickly initialize the dialog while running the packager at a later time. Also, as described below, the only way to define the download descriptor created by the command line version of the packager is to specify a download descriptor template file created using this dialog.

Rights and constraints

The rights dialog is used to define rights as defined in [DRMREL] when using the combined delivery or separate delivery DRM methods.

One can only define rights for content whose MIME type has compatible rights defined for it in the device type and only those rights defined. When using separate and combined delivery, the file list may only contain files of exactly one MIME type, so the application will know which types of rights, if any, are compatible with the content in the list, and the rights dialog will then display the rights defined as compatible with that MIME type. The list may not be empty because then the DRM Packager will have no idea for which MIME type one wants to define rights. At least one right must be enabled for the MIME type or the content will be unusable for the receiver of the content. The dialog will not accept input if this is not true.

The following example displays a rights dialog for a file list containing jpeg-files whose compatible rights are print and display:

At the top of the dialog, the user may define how the globally unique content id, connecting content to its rights by reference, is generated. The unique id for each rights file is generated automatically during the packaging process. The example label below the content id edit box displays in real-time how a content id would be generated.

By just choosing the 'Enable right' box without choosing the constraints (Count, Start time, End time, interval), an infite right will be generated.

To edit a right, one first selects the tab of that particular right. By using the checkboxes, one is able to enable or disable the chosen right as well as its constraints. The start time and end time constraints each has a button that upon activation brings up a calendar to enable the user to choose a date.

At the bottom of the dialog one can choose the rights encoding type, i.e. the format of the rights files, to be XML (*.dr files) or WBXML (*.drc files) as defined in [DRMREL].

It is also possible to define additional meta information about the content that will be incorporated into the protected content file (.dcf file, See [DRMCF] for more information) such as:

- Rights Issuer
- Content Name
- Content Description
- Content Vendor
- Icon URI

Note: The rights encoding type and meta information only applies to the separate delivery DRM method.

Rights template file

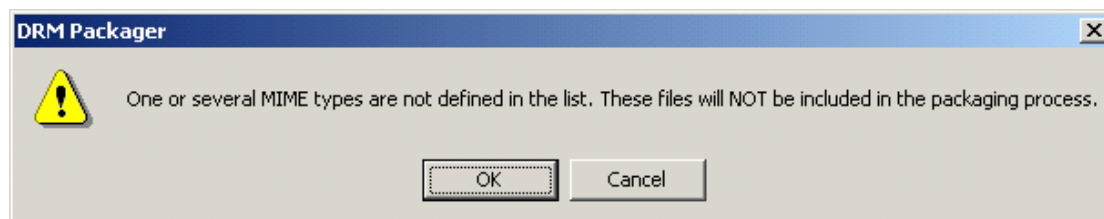
By using the "Load..."- and "Save..."-buttons in the dialog above, it is possible to load or save a rights template file. This type of file represents the information contained in the dialog in an XML format. It can be used to quickly initialize the dialog while running the packager at a later time. Also, the rights can be created using the command line version of the DRM Packager using the -x option. See "Command line version" on page 35 for more detailed information.

Packaging

When the list of content files to protect is complete and the user has chosen the configuration options desired, one can click the "Create DRM Content"-button or the "Create DRM Content"-menu alternative to begin the packaging process.

The list must contain AT LEAST one file with a MIME-type that is not empty, i.e. the MIME-type must be defined, otherwise the "Create DRM Content"-button and menu alternative will be deactivated and it will be impossible to start the packaging process.

If some files have an undefined MIME-type, these will be excluded from the packaging process automatically. In that case, the user will receive an alert message box as shown below where the user can choose to abort the packaging process to fill in the undefined MIME-types by pressing "Cancel"



If either of the DRM methods combined delivery and separate delivery are currently chosen, the application will check that at least one right is enabled for the MIME type of the content files in the list. If this is not true, the application will display the rights dialog.

If, during the creation of a protected content file, the tool discovers a file with the same path as that of the pending output file, it will ask the user to either cancel the entire operation, overwrite that single file or overwrite all such files.

Command line version

The command line version of this software is an executable called "DRMCmdLine.exe". It facilitates integration of the DRM Packager with other software by way of scripting.

The command line version can be executed in two different modes.

1. To generate rights template, using the -x switch. See "Rights generation options" on page 35.
2. To create DRM content, similar to the GUI based version mentioned earlier.

Syntax

The command line syntax is:

DRMCmdLine.exe -p<profile> **-x** <list of constrains> (when generating rights template)

DRMCmdLine.exe -f<input file> **-p**<profile> [further options] (when creating DRM content)

Note: It is advised to have the directory of the command line executable defined in the Windows/DOS PATH environment.

Command line parameters can be specified in any order. If multiple instances of the same parameter are provided, the application will only examine the first (leftmost) one. Non-existing command line options will be ignored. If a certain command line option is not provided the default value for that option will be used. If a parameter for some reason does not apply, such as not being compatible with the current DRM method, it is also ignored.

When parameters contain spaces, one must use quotation marks. Example:

DRMCmdLine.exe -f"content\my image name with spaces in it.gif" -pProfile\main.dpr.

Rights generation options

Rights can be associated to a DRM content using two methods:

1. By defining and saving a rights template in the Rights dialogue (See "Rights and constraints" on page 32)
2. Directly via the command line version using the -x switch.

When using the constrains associated with the rights, the constrains must be in the form <count>:<start>:<end>:<interval> where <start> and <end> are given in the form YYYYMMDDHHMMSS. If some of the constrains options are not meant to be used, then just leave these fields/values empty (e.g. <count>:::).

Constrains:

- file : Set output file. Default is RightsTemplate.xml in current directory.
- cid : Content ID to provide in template file. Example: --cid my-text@site.com
- xml : Output in XML format (default is WBXML).
- url : Rights Issuer URL. Example: --url <http://www.sonyericsson.com/>
- name : Content name. Example: --name Beach
- description : Content description. Example: --description Background
- vendor : Content vendor. Example: --vendor SonyEricsson

--*iconuri* : Icon URI (please observe that it is URI and not URL).

Example: --iconuri <http://www.sonyericsson.com/icon.ico>

--*display* : Set display rights.

--*print* : Set print rights.

--*play* : Set play rights.

--*execute* : Set execute rights.

Example: --display 2:20040416163400:20040417120000:20325123025 --play 22:::

Common command-line options

Flag	Mandatory	Default Value	Description
-h	No	OFF	Print help, e.g. the information in this list.
-q	No	OFF	Quiet mode. When used, no standard output terminal console is generated. The output is logged though.
-p	Yes		Path to profile file. Example: -pProfile\main.dpr.
-d	No	Current working directory	Output protected content directory. Ignored if option -s is used. Example: -dDRM.
-l	No	LogFile.txt	Path to log file. Example: -lLogFile.txt.

Command-line options for all DRM methods

Flag	Mandatory	Default Value	Description
-f	Yes. (Not used in rights generation mode -x)		Path to input file. Example: -fContent\Pic.gif. Example using spaces: -f"Content\My own img.gif"
-t	No	First device type in profile	Name of device type to use. Must be existent in profile. Example: -t"Sony Ericsson".
-r	No	flock	DRM method. Supported values: flock (forward lock), SonyEricsson, comb (combined delivery) and sep (separate delivery). Example: -rSonyEricsson.
-i	No		Path to Download Descriptor (DD) template file Example: -iProfile\DDTemplate.xml.
-s	No	No	Put output in subfolder named "DRM" in the folder of the input file. If this option is used then option -d will be ignored. Example: -s.
-o	No	No	If any output file (rights, download descriptor, content etc) with the same path as output exists overwrite it. Otherwise quit with an error message. Example: -o.

Command-line options for forward-lock only

-m	No	Lookup file extension of input file in profile and get MIME-type	MIME type for output file. Example: -mimage/gif.
----	----	--	---

Command-line options for forward-lock and combined delivery only

Flag	Mandatory	Default Value	Description
-e	No	binary	Encoding method. Must be supported by software and chosen device type. Supported types: Binary, 7bit, 8bit and base64. Example: -ebinary.

-n	No	Do not embed filename	Embed input filename. Embed filename into MIME body part headers. Example: -nmyname.gif. Example: -n. The last example lets the DRM Packager use the input file name as the embedded name.
-u	No	System default boundary	User defined MIME body part boundary. Example: -umyyboundary.

Command line options for combined and separate delivery only

Flag	Mandatory	Default Value	Description
-c	No	Use content id specification provided in the rights template file	Override content id specification in rights template file. Example: -cmy-cid-\$id@mysite.com.
-g	Yes		Path to rights template file. This is the save XML template file containing rights information generated in the DRM Packager (Rights dialog). Example: -gProfile\ RightsTemplate.xml.
-x			Generate a rights template file. See “Rights generation options” on page 35
-y	No	Current working directory	Rights output directory. Ignored if option -z is used. Example: -dMyRights.
-z	No	No	Put output in subfolder named "Rights" in the folder of the input file. If this option is used then option -y will be ignored. Example: -z.

Command line options for separate delivery only

Flag	Mandatory	Default Value	Description
-w	No	Use rights encoding type specified in rights template file	Override rights encoding type in rights template file. Supported types: xml, wbxml. Example: -wwbxml.

Device profile

Below is a small example of a device profile with all the necessary information for one device type. Note that the device profile is written in an XML format:

```
<profile>

<profile_name>Sony Ericsson, April 2004</profile_name>
<profile_version>1.2</profile_version>
<min_drm_tool_version>1.1</min_drm_tool_version>
<device_type>
  <name>Sony Ericsson</name>
    <drm_method>forward-lock</drm_method>
    <drm_method>combined delivery</drm_method>
    <drm_method>separate delivery</drm_method>
  <drm_method>Sony Ericsson</drm_method>
  <!-- allowed values: "forward-lock", "combined delivery", "separate delivery", "Sony Ericsson" -->
  <transfer_encoding>binary</transfer_encoding>
  <transfer_encoding>7bit</transfer_encoding>
  <transfer_encoding>8bit</transfer_encoding>
  <transfer_encoding>base64</transfer_encoding>
  <media_type>
    <mime_name>application/vnd.eri.thm</mime_name>
    <file_extension>thm</file_extension>
    <compatible_rights>
      <display/>
    </compatible_rights>
  </media_type>
  <media_type>
    <mime_name>image/jpeg</mime_name>
    <file_extension>jpg</file_extension>
    <file_extension>jpeg</file_extension>
    <compatible_rights>
      <display/>
      <print/>
    </compatible_rights>
  </media_type>
</device_type>
</profile>
```

The configuration dialog box will show the DRM methods and encoding types in their corresponding drop down lists in the same order as defined in the device profile. Also, by default, when a new device type is chosen either by using the drop down list or by loading a new device profile in the application, the first DRM method and encoding type will be chosen by the application automatically. This means that the above example profile would implicitly make the forward-lock method and the binary encoding type be chosen by default by the system should the user not make a manual choice.

According to [DRM] a device supporting combined delivery MUST support forward-lock and a device supporting separate delivery must support combined delivery. One may NOT enter combinations of DRM methods into the device profile that violate these constraints. For these reasons one should not change the order of the DRM methods supported in the example stated above for the application to work properly.

To define in the profile which rights, if any, are compatible with a certain MIME type, indicating that the MIME type is compatible with DRM methods combined delivery and separate delivery, one defines the tag "compatible_rights" as was done in the stated example for jpeg-files. In this example we can see that jpeg files can be displayed or printed but not played or executed. For thm-files no rights can be defined and thus that MIME type is incompatible with combined delivery and separate delivery.

There are two ways to add profile information to the application:

- 1) To add another device type, simply copy the text from the <device_type>-tag to the </device_type>-tag and paste it after the last <device_type>-tag, then alter the contents accordingly to add or remove DRM-methods, encoding types, supported media types and compatible rights.
- 2) To create another device profile (".dpr"-file), simply copy the original profile "main.dpr" and edit the copy. To use that profile, start the application and then select "Open device profile ..." from the "File"-menu.

IMPORTANT NOTES:

Method 2) above is the recommended practice. During installation/uninstallation, the default device profile (main.dpr) is deleted and will be replaced by the new default device profile.

It is important that the listed DRM methods and encoding types are supported by the DRM Packager currently in use.

A MIME type (<media_type>) can be associated with several file extensions as in the image/jpeg-MIME-type defined in the profile example above, which is associated to both .jpg and .jpeg.

MIME types

The following is a list of DRM related MIME-types:

file extension	MIME type	
.dr	application/ vnd.oma.drm.rights+xml	OMA DRM right in XML format
.drc	application/ vnd.oma.drm.rights+wbxml	OMA DRM right in WBXML format
.dm	application/ vnd.oma.drm.message	Forward-locked or Combined delivery content file
.dcf	application/ vnd.oma.drm.content	Separate delivery encrypted content file
.copy	application/ vnd.sem.mms.protected	Sony Ericsson proprietary protected content file

Index

B		T	
Billing systems	13	Text labels	28
Buttons.	29		
C		U	
Combined delivery	11, 20	User environment	12
Configuration options	25		
D			
Device profile	27		
Device type.....	27		
Download descriptors.....	12, 15		
Downloading servers	12		
DRM method.....	22		
DRM Package	r10, 25		
Dynamic web server	24		
E			
Edit menu.....	30		
F			
File list.....	28		
File menu	30		
Forward lock	17		
H			
HTTP-Response	17, 20		
I			
Input file list.....	25		
M			
Message Format	17		
Message format	20, 22		
MIDI standar	d7		
MMS Format.....	19		
O			
Output files.....	22		
Output path edit box.....	29		
S			
Separate delivery	11		
Server configuration.....	23		
Static web server	23		
Subscription.....	10		
Super distribution	9		