

In Confidence

**Connecting to Telenor Mobil's**

**Content Provider Access MMS (CPA  
MMS)**

**or**

**MMS shortnumber as additional  
service for SMS Bedrift and SMS  
Access**



**MCPA Server  
release 2.1**

**In Confidence**

Version 2.1

June 29 2009

# Table of Contents

<b>INTRODUCTION</b> .....	<b>8</b>
<b>CONTACTS IN TELENOR MOBIL</b> .....	<b>10</b>
SUPPORT DURING ORDER AND DELIVERY .....	10
SUPPORT AFTER DELIVERY .....	10
SALES CONTACT .....	10
<b>PREREQUISITES</b> .....	<b>11</b>
GETTING A VAS ACCOUNT .....	11
SOFTWARE INSTALLATION .....	11
CLASS-PATH .....	12
MIME.TYPES.....	12
BILLING INFORMATION .....	13
NORWEGIAN CHARACTERS AND CHARACTER ENCODING .....	14
<b>TESTING YOUR CONNECTION WITH SENDMMS</b> .....	<b>15</b>
<b>PROGRAMMING WITH THE API</b> .....	<b>16</b>
SENDING MESSAGES .....	16
RECEIVING MESSAGES AND DELIVERY REPORTS .....	16
IMPLEMENTING HTTP GET FOR CONNECTION TESTING.....	17
LOGGING.....	18
<b>MCPA RESPONSE CODES</b> .....	<b>19</b>
<b>SUPPORTED BILLING/SUBMISSION SCENARIOS</b> .....	<b>23</b>
ADDRESSING .....	23
BILLING FOR CPA MMS .....	23
BILLING FOR MMS ACCESS .....	23
<b>FUTURE FUNCTIONALITY</b> .....	<b>24</b>
DIGITAL RIGHTS MANAGEMENT .....	24
<b>HINTS</b> .....	<b>25</b>
<b>REFERENCES</b> .....	<b>26</b>
<b>APPENDIX A – USING BACKWARDS-COMPATIBLE API</b> .....	<b>27</b>
API FROM APRIL 2004 .....	27
API FROM MAY 2006 .....	28
<b>APPENDIX B –HINTS FOR MMS APPLICATIONS</b> .....	<b>29</b>
VERIFY YOUR BILLING.....	29
TEST YOUR CONTENT .....	29
BE USER-FRIENDLY .....	29
PERFORMANCE HINT .....	30
<b>APPENDIX C – CONFIGURING YOUR TERMINAL</b> .....	<b>31</b>
PRODUCTION MCPA .....	31
TEST MCPA.....	31
<b>APPENDIX D – SOAP IMPLEMENTATION GUIDE</b> .....	<b>33</b>

IMPLEMENTED MM7 MESSAGES .....	33
UNSUPPORTED MM7 MESSAGES.....	36
TRANSFER-ENCODED RESPONSE .....	36
CONNECTION TESTING .....	36

## Table of Figures

Figure 1 – First generation MCPA .....	8
Figure 2 – Second generation MCPA .....	9

## Definitions

Javadoc	Java source code documentation, available in HTML format, bundled with the API distribution.
Last resort	When a MM expires, the MMSC forwards the message to a “last resort” service that ensures that the subscriber receives it. This is a web-based e-mail service with SMS notification.
MMS Access	An MMS bulk product, where the VAS provider always pays for the submission. This is different from the CPA MMS product, where the subscribers pay for the submissions. MMS Access is sold as additional services to SMS Bedrift and SMS Access with the same short number on both SMS and MMS. SMS Bedrift is a product for business customers and SMS Access is a product for partners who develop SMS and MMS services for other customers on their own access / shortnumber.
SMS Access	
SMS Bedrift	
Service provider	A service provider in Telenor Mobil’s network resells mobile services using its own brand.
Subscriber	End-user of the value-added service.
System	Used as shorthand for “MCPA”, “CPA MMS middleware” or “MMS billing middleware”, i.e. the system described in this document.
VAS account	An account on MCPA with a user name and password, used to send and receive messages. A VAS provider may have one or more VAS accounts.
VAS provider	The term covers both Content Provider Access (CPA) and MMS Access.

## Abbreviations

API	Application Programming Interface; a software library that provides a programmer with access to some system.
CP	Content provider.
CPA	Content Provider Access. A group of systems that allows content providers to distribute content to subscribers, and bill the subscriber for the submission. CPA systems exist for SMS, MMS and WAP.
CPSC	Content Provider Subscription Centre. The CPSC server implements a billing concept where the subscriber pays once for access to a service for either a given number of times, or for a given time period.
DR	Delivery report.
DRM	Digital Rights Management. A system that enforces rules upon message submissions, so that it is possible to control the distribution of copyrighted content.
DTR	“Datatjenester, rådgivning og leveranse”. The CPA MMS and MMS Access support team.
HTML	Hypertext Markup Language. Used to write human-readable web pages.
HTTP	Hypertext Transfer Protocol. Typically used to request a HTML document, or transfer SOAP messages.
MCPA	<i>MMS Content Provider Access</i> . Shorthand notation for CPA MMS – the MMS gateway for content providers. This is the most common name of the server, although it runs both the MMS Access and CPA MMS services.
MM	MMS Message or Multimedia Message.
MM7	Content provider interface defined in the MMS standard [ETSI123v5]. It uses SOAP over HTTP as message transfer protocol.
MMS	Multimedia Messaging Service.
MMSC	Multimedia Messaging Service Centre. MMS node in the mobile network.
MO	Mobile Originated (e.g. MO MMS is a MM which is sent <i>from</i> the mobile terminal)
MSISDN	Mobile phone number, including international prefix. Eight digits in Norway.
MT	Mobile Terminated (e.g. MT MMS is a MM which is sent <i>to</i> the mobile terminal)
O&M	Operations and maintenance. This is a term used for managing the system in its production environment.
SOAP	Simple Object Access Protocol. An Internet-standards-based remote procedure call mechanism.
TnM	Company-internal shorthand notation for Telenor Mobil.

URL	Internet address.
VAS	Value Added Service. E.g. news, weather, ringtones, etc.
VASP	VAS Provider – provider of value added services, i.e. either a CP or a MMS Access provider/customer.

## Introduction

This document contains information required for VAS providers to send and receive multimedia messages using Telenor Mobil's CPA MMS server (MCPA). It is part of a distribution bundle, also containing the MCPA Java API and API documentation [MCPAAPI]. The target audience is developers.

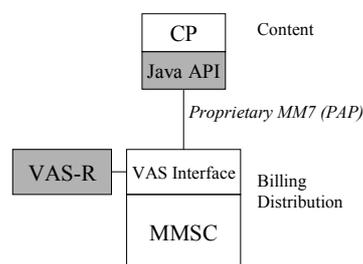
The MCPA server supports two business models, CPA MMS and MMS Access. The first provides reverse charging, where the recipients pay for the distributed content, while the latter is a bulk distribution service, where the VASP sender pays. MMS Access is sold as an additional service to SMS Bedrift and SMS Access with the same shortnumber for both SMS and MMS.

The MCPA server implements the VAS interface specified in the MMS functional specification [ETSI123v5], also referred to as *MM7*. All mandatory elements in the MM7 specification are implemented. In brief, MCPA supports the following features:

- Sending multimedia messages to one or more subscribers. The recipient may belong to any operator with which Telenor Mobil has MMS roaming. The payer may be either sender or recipient, but must be a Telenor Mobil subscriber, or a subscriber of any of Telenor Mobil's service providers. The payer may also be a VAS provider.
- Delivery reports are supported.
- Receiving mobile originated messages. The sender may be a Telenor Mobil subscriber, or a subscriber of any of Telenor Mobil's service providers. There is no premium rate for mobile originated messages. The subscriber pays the same price as when sending to other persons.

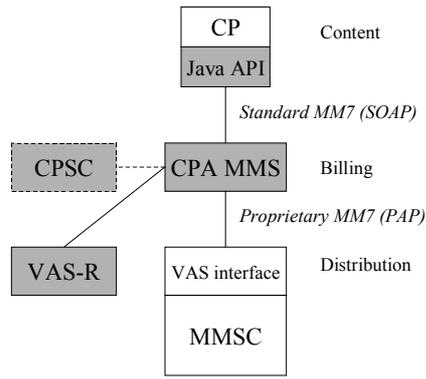
For details on implemented features, please refer to "Appendix D – SOAP Implementation Guide".

This is the second generation MCPA in Telenor Mobil. The previous version consisted of a Java client library (API) communicating with the MMSC over the proprietary PAP-based MMSC VAS interface [MCPAPAP]. The MMSC node handled both traffic handling and billing.



**Figure 1 – First generation MCPA**

This second generation MCPA implements the billing functionality in a layer separate from the traffic handling. The Java API is totally re-written. However, (nearly) backwards-compatible classes are available for those not willing to migrate to the new API – thereby missing out on new features like delivery reports and recipient charging. The API communicates with MCPA over the standard MM7 SOAP interface, which is specified in the MMS specification.



**Figure 2 – Second generation MCPA**

This document only describes usage of the new, re-written API. Please refer to the Javadoc [MCPAAPI] and “Appendix A – Using Backwards-Compatible API” for information about the old classes.

## Contacts in Telenor Mobil

This section will assist you in contacting the right people in Telenor.

### ***Support During Order and Delivery***

To order a VAS account, contact the Partner Solution support team at

[Mobil.data@telenor.com](mailto:Mobil.data@telenor.com). The latest version of API and documentation is always available on

<http://cpa.telenor.no>. The support team will provide you with any passwords you may require

to access the site.

All questions during the order and delivery process should be directed to the Partner Solution support team. This includes API programming questions, VAS account changes, general MCPA questions, and error situations.

Any questions regarding the test system must be directed to the Partner Solution support team.

### ***Support After Delivery***

After going live with your service, the order and delivery process is complete, and you must direct any questions about your MCPA connection to the helpdesk *Nettfront*, (see agreement) or e-mail (see agreement) This applies for downtime, or if the server starts responding in an unexpected manner.

Do not contact Nettfront with questions regarding the test system; use the Partner Solution support team instead.

Nettfront distributes error notifications per SMS and e-mail. Contact them if you wish to subscribe on one of these distribution lists.

### ***Sales contact***

Shortnumber for MMS is sold at an additional service to the SMS Bedrift and SMS Access service

Sales contracts and commercial questions shall be directed to your sales contact in Telenor.

## Prerequisites

Before you start programming, all the requirements in this chapter must be fulfilled.

### Getting a VAS Account

By the time you are reading this, you will already have obtained a CPA MMS or MMS Access account on either the test or production MCPA server, or both. If not, order one by contacting the Partner Solution support team at [mobil.data@telenor.com](mailto:mobil.data@telenor.com). This process includes filling in a customer information form.

You will then receive the MCPA server's URL, and username and password for the connection. Please note that the MCPA URL is different than the URL to the MMSC, if you have been sending MMS messages previously.

### Software Installation

To get the API up and running, you first have to obtain the MCPA client distribution bundle. It may be downloaded from <http://cpa.telenor.no>.

The distribution contains both documentation and Java API classes. Other execution environments, like .NET is currently not supported. Please refer to "Appendix D – SOAP Implementation Guide" to implement your own client.

After unzipping the distribution into a directory, the bundle has the following structure:

/bin	
sendmms.bat	Test application.
log4j.xml	Log configuration for sendmms application.
log4j.dtd	
testmessage.zip	An MMS compatible with most terminals.
mime.types	Copy mime.types to your JRE's lib-folder, if you do not already have one such file that is working for your MMS service.
/lib	
All required jar-files	
/doc	
MCPAGettingStarted.doc	This document.
/doc/api	
Javadoc	Detailed MCPA client API documentation [MCPAAPI]

## Class-Path

The following Java archives (JAR) must be in your application's class-path. They should reside in the same folder, as they reference each other by name.

- MCPASOAP.jar – the MCPA client API. These are the classes you use in your application. They are documented in the /doc/api folder of the MCPA distribution.
- mm7common.jar – classes that the client API use to communicate with MCPA over the SOAP interface.
- activation.jar – classes used for mapping file extension to content type.
- commons-logging.jar – Jakarta log framework used by dom4j. Note that if you are using the Weblogic application server, it may require log4j-core.jar to be in your class-path as well, as the commons-logging jar-file references the log4j-core library in its manifest file, and Weblogic checks these references.
- dom4j.jar – XML toolkit.
- log4j.jar – log framework.
- mail.jar – used for MIME building and parsing.
- saaj-api.jar – SOAP API interfaces.
- saaj-ri.jar – SOAP API implementation.
- xml-apis.jar – a couple of low-level XML classes used for efficient parsing.

In addition, test.jar contains an application with which you may test your connection. Read more in the chapter “Testing Your Connection”.

## *mime.types*

The Activation framework (activation.jar), expects to find the file *mime.types* in your Java Runtime's lib-folder, e.g. `c:/j2sdk1.4.1/lib`. If this file is not found, the mapping from file extension to content type will not work correctly, and the content will not display properly on the terminals. If the content is not displayed according to your SMIL document, then you should check that *mime.types* is placed correctly.

You should also verify that the file contains mappings for the file extensions you use. Below are some common mappings.

application/smil	smil
audio/AMR	amr
audio/imelody	imy
audio/midi	mid midi
text/plain	txt
image/gif	gif
image/jpeg	jpeg jpg

If you do not use file extensions, the class `MMSContentFactory` provides methods that allow you to set the content type programmatically.

### MIME types file search order:

The `MimetypesFileTypeMap` looks in various places in the user's system for MIME types file entries. When requests are made to search for MIME types in the `MimetypesFileTypeMap`, it searches MIME types files in the following order:

1. Programmatically added entries to the `MimetypesFileTypeMap` instance.
2. The file `.mime.types` in the user's home directory.
3. The file `<java.home>/lib/mime.types`.
4. The file or resources named `META-INF/mime.types`.
5. The file or resource named `META-INF/mimetypes.default` (usually found only in the `activation.jar` file).

Read more about it here:

[http://java.sun.com/j2ee/1.4/docs/api/javax/activation/MimetypesFileTypeMap.html#getContentType\(java.lang.String\)](http://java.sun.com/j2ee/1.4/docs/api/javax/activation/MimetypesFileTypeMap.html#getContentType(java.lang.String))

## Billing Information

When submitting, you indicate whether the sender or recipient shall pay. The price is part of the billing information, which consists of three elements. The billing information is usually specified as a # separated string: `PriceCategory#ServiceID#ServiceCategory`, e.g. `'CPAMMS1000#News#AnyText'`

### CPA MMS

The following billing information guidelines apply for CPA MMS.

PriceCategory	10 characters CPAMMSXXXX	XXXX is 1/100 NOK (=øre). Example: CPAMMS1000 is 10 NOK.
ServiceID	Max 6 characters	Your product identifier, which is used when Telenor Mobil generates statistics. The value may be any string of your choice. ServiceID will also appear on the subscriber's invoice, combined with your VAS account's short number, e.g. "News:1999".
ServiceCategory		Reserved for categories defined by Telenor. It is currently not in use, so you may choose any string shorter than 30 characters, e.g. "NotUsed".

CPA MMS product management will provide you with updated information about valid price categories and service categories. As of December 2003, valid price categories for CPA MMS are CPAMMS0000, and CPAMMS0300 – CPAMMS3000 (i.e. 0 NOK, and 3 – 30 NOK),

plus categories for 30, 35, 39, 40, 45, 49, 50, 55, 59 and 60 NOK. CPAMMS0000 is a special price category, where you as a content provider are charged an amount for the submission.

## MMS Access

The following billing information guidelines apply for MMS Access.

PriceCategory	CPAMMS0000	You must use this price category.
ServiceID	Max 6 characters	Your product identifier, which is used when Telenor Mobil generates statistics. The value may be any string of your choice.
ServiceCategory		Reserved for categories defined by Telenor. It is currently not in use, so you may choose any string shorter than 30 characters, e.g. "NotUsed".

## Norwegian Characters and Character Encoding

If you experience problems with sending and receiving messages with Norwegian characters (æ, ø and å), try changing your system's default character encoding to ISO-8859.

Microsoft Windows systems have ISO-8859-1 as standard character encoding, so this should mostly apply for Unix systems.

## Testing Your Connection with SendMMS

The distribution bundle contains an application, SendMMS, which allows you to send messages, without building your own application. The application is command-line-driven. It supports sending to both single and multiple recipients, and charging either sender or recipient.

First, modify the following lines in `/bin/sendmms.bat` to match your VAS account settings.

```
set MMSC=http://web-access.mobil.telenor.no/mcpa/vaspreceiver
set USERNAME=aUsername
set PASSWORD=aPassword
set BILLING=CPAMMS0000#TestID#TestDescription
set RECEIVERPAYS=true
set ORDERDELIVERYREPORT=false
```

*Note that the parameter MMSC is pointing to the production MCPA. For testing towards the test MCPA use the following url:*

*set MMSC= http://mcpa-pilot.mobil.telenor.no/vaspreceiver*

The command line syntax is as follows:

```
sendmms <sender> <recipient> <title> <path-to-mms>
```

`<path-to-mms>` may be either a folder containing your content files, or the name of a ZIP-file with the content files. Absolute and relative addressing is supported. The content in folder or ZIP-file must only contain the files to be sent, without any subfolders.

`<recipient>` may be either a phone number, a Static ID, or the name of a text file with one phone number per line.

**NOTE: Duplicate recipients will be removed.**

An example, that uses the test message in `/bin/testmessage.zip`:

```
sendmms 1234 99999991 "My test" testmessage.zip
```

Execute “sendmms” without any parameters from the `/bin`-folder, to get help with the more advanced features.

The current SendMMS startup script only supports Windows. Unix users should be able to modify the script to run on their system as well.

## Programming with the API

The Javadoc distributed with the API is your main source of programming information [MCPAAPI]. This chapter will get you started. Refer to the Javadoc for details.

### *Sending Messages*

To submit a message, you need a sender object.

```
MMSsender sender = new SOAPSender(mmscURL, username, password);
```

Then, use MMSContentFactory to create the content element objects. You may create content from inputstreams, files in folders, or files in ZIP archives.

```
MMSContentElement[] contentElements =  
MMSContentFactory.createFromFolder("c:/path-to-my-mms-folder");
```

Finally, you create the message, set optional parameters, and submit it.

```
SubmitMessage message = new SubmitMessage(originator, receivers, receiverPays,  
    priceCategory, serviceID, serviceCategory);  
message.setContent(contentElements);  
message.setOrderDeliveryReport(true); // optional  
SendingResponse response = sender.submit(message);
```

After sending, you may retrieve submission status from the SendingResponse object.

```
System.out.println("MCPA returned code " + response.getResultCode() + " - "  
    + response.getResultDescription());
```

Please refer to the Javadoc [MCPAAPI] for details on SOAPSender and the other involved classes.

### *Receiving Messages and Delivery Reports*

You need a Java servlet to receive messages and delivery reports from the MCPA. Once you have received a message from MCPA, you check whether it is a mobile originated message (also called ‘deliver’), or a delivery report.

```
public void doPost(HttpServletRequest httpRequest,  
    HttpServletResponse httpServletResponse){  
  
    SOAPReceiver soapReceiver = new SOAPReceiver();  
    try {  
        ReceiveMessage inMessage = soapReceiver.receive(request, response);  
        If (inMessage.isDeliver()){  
            DeliverMessage deliverMessage = (DeliverMessage) inMessage;  
            String originator = messageFromSubscriber.getSender();  
            String[] recipients = messageFromSubscriber.getRecipients();  
            String subject = messageFromSubscriber.getSubject();  
            MMSContentElement[] contentElements =
```

```
        messageFromSubscriber.getContent();

    } else if(inMessage.isDeliveryReport()){
        DeliveryReportMessage deliveryReport =
            (DeliveryReportMessage) inMessage;
        String senderOfOriginalMessage = deliveryReport.getSender();
        String recipientOfOriginalMessage = deliveryReport.getRecipient();
        String deliveryStatus = deliveryReport.getStatus();
        String messageIdOfOriginalMessage = deliveryReport.getMessageID();
    } else {
        System.out.println("Got unknown messagetype from MCPA");
    }
} catch (MCPAException me){
    System.out.println("Error receiving message : ");
    me.printStackTrace();
}
}
```

Please refer to the Javadoc [MCPAAPI] for details on SOAPReceiver, DeliveryStatus, and the other involved classes.

## ***Implementing HTTP GET for Connection Testing***

MCPA implements a proprietary scheme to obtain robustness against VAS providers. This requires that the VAS application accept HTTP GET requests.

If a message delivery or a delivery report to a VAS provider fails, then MCPA will flag this connection as ‘down’ and stop delivering more messages to the VAS provider. At the same time, MCPA starts to periodically connect to the VAS application’s URL. As soon as this connection succeeds, the connection is flagged as ‘up’, and MCPA will start to deliver messages again. This is done to reduce the risk of time-out situations due to network problems or erroneous VAS applications.

In order to make this work, the VAS application must respond to HTTP GET requests on the same URL as the messages are posted. An example for your Java Servlet is provided below.

```
public void doGet(HttpServletRequest request, HttpServletResponse response)
    throws IOException {

    response.setContentType("text/plain");
    response.getOutputStream().println("HTTP GET not supported for this URL");

}
```

This will result in a successful connect to the URL, which MCPA interprets as a reachable VAS application.

## Logging

The API uses Log4J log framework [Log4J]. Two files must be present for logging to work.

- log4j.xml. Log configuration.
- log4j.dtd. Document type definition for log4j.xml. Do not alter this file.

If your application uses Log4J, then this configuration will apply also for the MCPA API. If not, you may put the two files in your classpath. If no configuration is available, then no logging will occur. The API will work even without logging.

The following *categories* are most relevant for you to tune in log4j.xml, in order to get the logging information you desire.

- com.telenor.mcpaclient. This is the new MCPA API. Recommended log level is INFO.
- com.telenor.mcpa. This is the backwards compatible MCPA API. Recommended log level is INFO.
- com.telenor.mcpaclient.test. This package contains the SendMMS test application bundled with this API. Recommended log level is INFO.
- com.telenor.mcpacommon. This package contains classes that implement the MM7 SOAP protocol. It is used by the packages com.telenor.mcpaclient and com.telenor.mcpa. Recommended log level is WARN.

NOTE: You should never program directly against any classes in this package, so try to avoid putting them in your programming editor's class-path.

Please refer to the Log4J documentation for further details on how to configure logging.

## MCPA Response Codes

Below is a list of response codes, with troubleshooting help. *The list is based on the MMS specification, and not all codes will be returned by the MCPA server.* Codes not returned by the current MCPA version is written in grey.

The MCPA server's response will also contain a "details"-field. You should always check the error details for hints on why your submission failed.

Please note that these codes only apply when using the new API in MCPASOAP.jar. When using the classes that are backwards compatible with MCPAPAP, the response codes are the old ones, described in the MCPAPAP documentation [MCPAPAP].

Status Code	StatusText	Interpretation
1000	Success	<p>MMS Specification: This code indicates that the request was executed completely.</p> <p>MCPA MM7_submit: MCPA accepted the message, and will attempt to deliver it to the recipient(s).</p>
1100	Partial success	<p>MMS Specification: This code indicates that the request was executed partially but some parts of the request could not be completed. Lower order digits and the optional Details element may indicate what parts of the request were not completed.</p> <p>MCPA MM7_submit: MCPA accepted some, but not all recipients. The details field will provide information about failed recipients. Reasons for rejection may be an illegal number or that the recipient has insufficient prepaid credit.</p>
2000	Client error	<p>MMS Specification: Client made an invalid request.</p> <p>MCPA MM7_submit: This error code may arise from a number of reasons:</p> <ul style="list-style-type: none"> <li>• Too many recipients. Reduce the number of recipients per submission, and try again.</li> <li>• Wrong rate or impact category. This may indicate that the VAS account is not correctly configured.</li> <li>• Invoice text is too long. This typically means that the VASP has too many characters in the billing information string, more specifically the ServiceCode.</li> <li>• Unsupported charged party. The VASP has in some way specified another recipient than sender or recipient.</li> </ul>

2001	Operation restricted	<p>MMS Specification: The request was refused due to lack of permission to execute the command.</p> <p>MCPA MM7_submit: This error code may arise from a number of reasons:</p> <ul style="list-style-type: none"> <li>• Authentication failed. Wrong username and password.</li> <li>• Sender address not accepted. If recipients pay, then the VAS short number is the only allowed sender address.</li> <li>• Illegal price category in billing information string.</li> <li>• VAS account is blocked, and is thus not allowed to send messages.</li> <li>• VAS account is blocked, and is thus not allowed to send messages to external recipients with pricecategory CPAMMS0000.</li> </ul>
2002	Address Error	<p>MMS Specification: The address supplied in the request was not in a recognized format or the MMS Relay/Server ascertained that the address was not valid for the network because it was determined not to be serviced by this MMS Relay/Server. When used in response-result, and multiple recipients were specified in the corresponding push submission, this status code indicates that at least one address is incorrect.</p> <p>MCPA MM7_submit: This error code may arise from a number of reasons:</p> <ul style="list-style-type: none"> <li>• All recipients failed. No recipients were accepted.</li> <li>• Unknown address format.</li> </ul>
2003	Address Not Found	<p>MMS Specification: The address supplied in the request could not be located by the MMS Relay/Server. This code is returned when an operation is requested on a previously submitted message and the MMS Relay/Server cannot find the message for the address specified.</p> <p>MCPA MM7_submit: This error will occur when MCPA relies on interconnected MMSC servers to deliver the message in another network, but no such interconnection exist for the recipient number.</p>
2004	Multimedia content refused	<p>MMS Specification: The server could not parse the MIME content that was attached to the SOAP message and indicated by the Content element or the content size or media type was unacceptable.</p> <p>MCPA MM7_submit: Message size exceeds the size limit for this VAS account.</p>
2005	Message ID Not	<p>MMS Specification:</p>

	found	<p>This code is returned when an operation is requested on a previously submitted message and the MMS Relay/Server cannot find the message for the message ID specified or when the VASP receives a report concerning a previously submitted message and the message ID is not recognized.</p> <p>MCPA: Only used for MM7_deliver and MM7_delivery_report.</p>
2006	LinkedID not found	<p>This code is returned when a LinkedID was supplied and the MMS Relay/Server could not find the related message.</p>
2007	Message format corrupt	<p>MMS Specification: An element value format is inappropriate or incorrect.</p> <p>MCPA MM7_submit: Could not parse billing information string.</p>
3000	Server Error	<p>The server failed to fulfill an apparently valid request.</p> <p>MCPA MM7_submit: This error code may arise from a number of reasons:</p> <ul style="list-style-type: none"> <li>• An internal server error occurred in MCPA. Please make a note of the details information in the submission response.</li> <li>• The MCPA billing subsystem experienced a critical error that made queuing of the billing request impossible.</li> <li>• MCPA experienced an error when attempting to submit the message to the MMSC. All recipients failed.</li> </ul>
3001	Not Possible	<p>The request could not be carried out because it is not possible. This code is normally used as a result of a cancel or status query on a message that is no longer available for cancel or status query. The MMS Relay/Server has recognized the message in question, but it cannot fulfill the request because the message is already complete or status is no longer available.</p>
3002	Message rejected	<p>MMS Specification: Server could not complete the service requested.</p> <p>MCPA: Only used for MM7_deliver and MM7_delivery_report.</p>
3003	Multiple addresses not supported	<p>The MMS Relay/Server does not support this operation on multiple recipients. The operation MAY be resubmitted as multiple single recipient operations.</p>
4000	General service error	<p>The requested service cannot be fulfilled.</p>
4001	Improper identification	<p>Identification header of the request does not uniquely identify the client (either the VASP or MMS Relay/Server).</p>
4002	Unsupported version	<p>The version indicated by the MM7 Version element is not supported.</p>

4003	Unsupported operation	<p>MMS Specification: The server does not support the request indicated by the MessageType element in the header of the message.</p> <p>MCPA: MCPA does not support this operation.</p>
4004	Validation error	<p>MMS Specification: The SOAP and XML structures could not be parsed, mandatory fields are missing, or the messageformat is not compatible to the format specified. Details field may specify the parsing error that caused this status.</p> <p>MCPA MM7_submit: This error code may arise from a number of reasons:</p> <ul style="list-style-type: none"> <li>• No transaction ID provided.</li> <li>• SOAP parsing failed.</li> </ul>
4005	Service error	<p>MMS Specification: The operation caused a server (either MMS Relay/Server or VASP) failure and should not be resent.</p> <p>MCPA: Only used for MM7_deliver and MM7_delivery_report.</p>
4006	Service unavailable	<p>MMS Specification: This indication may be sent by the server when service is temporarily unavailable, e.g. when server is busy.</p> <p>MCPA MM7_submit: The MMSC is currently unavailable. Please verify that the error has been registered by Telenor Mobil, and try again later.</p>
4007	Service denied	<p>MMS Specification: The client does not have permission or funds to perform the requested operation.</p> <p>MCPA MM7_submit: This error code may arise from a number of reasons:</p> <ul style="list-style-type: none"> <li>• No payers were accepted. When sending to single recipients, this means that prepaid credit was insufficient or the payer is not in Telenor Mobil's network.</li> <li>• VAS provider not allowed as payer. If submitting a CPA message, only subscribers are allowed as payers. If using VAS provider short code as sender address, then the recipient must be payer.</li> <li>• The payer is barred from CPA content and will not receive any MMS.</li> </ul>

## Supported Billing/Submission Scenarios

The following section describes supported combinations of *sender*, *receiver* and *payer* when submitting messages.

### **Addressing**

Multimedia messages may be sent to one or more recipients. The recipient may belong to any operator with which Telenor Mobil has MMS roaming.

If the subscriber belongs to a service provider in Telenor Mobil's network, then she may not be provisioned with the MMS service. In that case, MMS will not work, and she must contact her service provider's customer service, to be provisioned with MMS.

Norwegian numbers may be either eight digits (xxxxxxx) long, with country code (47xxxxxxx), or with international prefix and country code (+47xxxxxxx or 0047xxxxxxx). Foreign numbers must have international prefix and country code.

When sending to multiple recipients, the recipient addresses must be unique per submission. Duplicate recipient addresses will be removed.

### **Billing for CPA MMS**

For CPA MMS, the payer may be either sender or recipient, but must be a Telenor Mobil subscriber, or a subscriber of any of Telenor Mobil's service providers.

Submitting to a list of recipients, where the recipients pay is allowed. When the sender pays for a submission to multiple recipients, the payer is charged the price of the message, multiplied with the number of recipients.

It is allowed to submit with your VAS provider short number as from-address, provided that the recipient pays. Also note that most MCPA accounts are configured so that a subscriber's number in the from-field will be replaced with your short number.

The *time of billing* is worth some extra attention. *Prepaid* subscriptions are billed upon submission. But if a delivery report indicates that the message was not delivered, the subscriber is refunded. *Postpaid* subscribers are billed when a delivery report indicates that the message was successfully delivered. And submissions to recipients that are external to Telenor Mobil's MMSC will be billed upon submission, as no delivery reports are available. This billing procedure enables more precise billing, which is important to ensure that the end-users get a good experience from using your service.

### **Billing for MMS Access**

For MMS Access, there is only one billing scenario: The sender must always be defined as the payer, and the sender address must always be the five-digit VAS provider short number.

## Future Functionality

The following features are planned for future MCPA releases.

### ***Digital Rights Management***

MCPA will implement support for Digital Rights Management (DRM), when Telenor Mobil's MMSC supports this functionality. DRM support will probably be limited to the features supported by the standard MM7 interface [ETSI123v5].

## Hints

When using Weblogic 8.1 there may encounter ClassCastExceptions. Here are some hints for fixing them:

- Force Weblogic 8.1 to use another SoapMessage Factory then the one that comes with Weblogic 8.1. This is done by adding a Java system property in the command line that starts the Weblogic 8.1  
- Djavax.xml.soap.MessageFactory =  
com.sun.xml.messaging.saaj.soap.MessageFactoryImpl
- Isolate the web applications lib classes so that they are preferred during class loading and not the ones for Weblogic 8.1. This is done by adding, “prefer-web-inf-classes” to “true” in the web.xml file for the application. Also all the MCPA libs must be put under the applications WEB-INF/lib.
- To read more about ClassCastException see [http://support.bea.com/application\\_content/product\\_portlets/support\\_patterns/wls/Class\\_Cast\\_Exceptions\\_Pattern.html#Using\\_Prefer-web-inf-classes\\_feature](http://support.bea.com/application_content/product_portlets/support_patterns/wls/Class_Cast_Exceptions_Pattern.html#Using_Prefer-web-inf-classes_feature)

## References

These are the external references used in this document.

[ETSI123v5]	Multimedia Messaging Service, Functional Description, Stage 2, version 5.7.0. Document ID is 123.140.
[Log4J]	Log4J logging framework. <a href="http://jakarta.apache.org/log4j">http://jakarta.apache.org/log4j</a> (online)
[MCPAAPI]	Javadoc located in the /doc/api folder of the MCPA distribution.
[MCPAPAP]	First generation MCPA API with documentation. <a href="http://cpa.telenor.no">http://cpa.telenor.no</a> (online)
[MMSCONF]	MMS Conformance Document v1.1. <a href="http://www.ericsson.com/mobilityworld/developerszonedownload/downloads/docs/messaging/MMS_Conformance_version_1.pdf">http://www.ericsson.com/mobilityworld/developerszonedownload/downloads/docs/messaging/MMS_Conformance_version_1.pdf</a>

## Appendix A – Using Backwards-Compatible API

### *API from april 2004*

For those of you that need to postpone using the new MCPA API, the two old classes MMSMessage and MMSReceive are still available for submitting and receiving messages. They are almost backwards compatible, in the sense that some methods have been removed, since they were not possible to migrate between the PAP and SOAP interfaces. But the most widely used send methods are still available. Minor changes must be made for receiving messages.

Please refer to the Javadoc [MCPAAPI] for package com.telenor.mcpa.pap for details about programming with these classes. Their use is deprecated, and they will not be supported in future releases of the MCPA API. You should therefore migrate to the new API. This will enable you to receive delivery reports and charge recipients as well.

#### **Procedure for migration**

1. Replace the libraries that belong to the old MCPAPAP API (jcert.jar, jdom.jar, jnet.jar, jsse.jar, MCPAPAP.jar and xerces.jar) with the libraries that belong to the new MCPASOAP API (commons-logging.jar, dom4j.jar, log4j.jar, MCPASOAP.jar, mm7common.jar, saaj-api.jar, saaj-ri.jar, xml-apis.jar). Leave the libraries that are common for the two APIs (activation.jar, mail.jar). You should obviously leave any of the deprecated libraries, if your own code depends upon them, but be sure to remove the MCPAPAP.jar.
2. Modify your application's class-path, to include the new JAR files instead of the old ones.
3. Recompile your application. If you used any of the methods that are no longer supported, you will have to modify your code to make it compile. If your application do not receive multimedia messages, and you use send methods that are still supported, and you feel brave; then you may skip recompiling, and proceed to the next step in the migration procedure.
4. Change your MMSC URL to refer to MCPA instead. The MCPA support team ([dtr@telenor.com](mailto:dtr@telenor.com)) will provide you with your new settings. If your application receives messages, then please inform the support team of the new URL to your receiver. If migrating production traffic, then make sure that you migrate at the same time as Telenor Mobil reconfigures the URL. These two events must be synchronized.
5. Restart your application.

This procedure should be sufficient for a successful migration to the new MCPA server. If you want to use the logging feature of this API, then please refer to the section “Logging” for details on how to configure Log4J.

You should make sure that your Java Servlet responds to HTTP GET requests, as described in the section “Implementing HTTP GET for Connection Testing”.

Please note that the file mime.types still must be available to the Activation library. Refer to the section “mime.types” for details.

Also note that the response codes when submitting to MCPA have not changed from MCPAPAP to the backwards-compatible version of MCPASOAP.

## API from May 2006

*If you are not interested in the new features in this new API you don't have to do any changes in your application/MM7/Soap client.*

This new API supports content-id and content-transfer-encoding for MO messages. To obtain the new parameters, please refer to the Javadoc [MCPAAPI] for package com.telenor.mcpaclient.MMSContentElement for details about programming with the new methods getContentTransferEncoding() and getContentId(). If you are using your own MM7/Soap client the content-id and content-transfer-encoding will be found in the MM7 message from MCPA.

Here is a programming example:

```
MMSContentElement[] content = deliverMessage.getContent();
for (int i = 0; i < content.length; i++) {
    MMSContentElement contentElement = content[i];
    String contentid = content[i].getContentId();
    String contenttransferencoding=content[i].getContentTransferEncoding();
    File directory = new File(contentDir + File.separator + basename);
        if (!directory.exists()) {
            directory.mkdir();
        }

    String contentFilename = contentDir + File.separator + basename + File.separator
+ contentElement.getName();
    FileOutputStream fosp = new FileOutputStream(contentFilename);
    fosp.write(contentElement.getDataAsBytes());
    fosp.close();
}
```

### Procedure for migration

1. Replace the libraries that belong to the old MCPA API with the libraries that belong to the new MCPA API.
2. Change your application with the new get-methods.
3. Recompile your application.
4. Restart your application.

This procedure should be sufficient for a successful migration to the upgraded MCPA server.

This new API also support content-id for MT messages, but MCPA sets the content-id itself by using the filename, e.g. filename=Picture.jpg -> content-id=<Picture.jpg>

## Appendix B – Hints for MMS Applications

When developing MMS applications, there are a few guidelines that will help you develop a working application.

### ***Verify Your Billing***

Verify that billing works. Use your service with a prepaid subscription, and verify that the correct amount is withdrawn from the prepaid account. It is your responsibility that the billing information for each submission is correct.

### ***Test Your Content***

Make sure that your content works on every MMS terminal, or adapt your content to make use of the user's specific terminal. To make your content work on a maximum number of terminals, make sure that you conform to the MMS Conformance Document [MMSCONF]. However, experience has shown that some terminals add further restrictions to the guidelines in the conformance document.

During autumn 2003, the following guidelines proved to work on all available MMS terminals:

- Maximum image size should be 96 x 64. Text readability is best when anti-aliasing is used in GIF files. Use a maximum of 256 colours. Using the Browser Safe Palette (search the Internet) is recommended. Usually, GIF is best for logos, and JPG for pictures.
- If possible, use short filenames (8+3). This makes SMIL work on all terminals.
- Your SMIL file should look something like this (which conforms to the MMS Conformance Document):

```
<smil>
  <head>
    <layout>
      <root-layout width="160" height="140"/>
      <region id="Image" width="160" height="120" left="0" top="0"/>
      <region id="Text" width="160" height="20" left="0" top="120"/>
    </layout>
  </head>
  <body>
    <par dur="10s">
      
      <text src="text.txt" region="Text" />
    </par>
  </body>
</smil>
```

### ***Be User-Friendly***

Check the MMSC response you get when submitting messages, and use detailed error messages telling the user if something went wrong. If the submission failed, she should try

again later. If her prepaid credit is too low, she should recharge her account, etc. This helps the user complete her purchase of your service.

### ***Performance Hint***

If your recipient list is longer than the maximum allowed number of recipients per submission, e.g. 25, then you should partition your list into shorter lists, which you send sequentially or in parallel.

When sending to lists, maximum throughput is achieved by sending short lists of 10-15 recipients, and in parallel. Try to send to ten and ten recipients, with e.g. five simultaneous sender threads. This way, you gain maximum from parallelism in server-side processing.

## Appendix C – Configuring Your Terminal

Depending on which instance of the MCPA server you connect to, you need different MMS settings on your terminal.

Most terminals support multiple MMS settings, and enable you to easily switch between them. Please note that WAP 2.0-compliant terminals may communicate using TCP/IP instead of WAP. This affects your terminal settings.

### ***Production MCPA***

The production MCPA uses the production MMSC for sending and receiving messages, and you must therefore configure your terminal to use the production MMSC. Most terminals are already pre-configured at point of sale, but you can also set up your terminal at <http://telenormobil.no>.

If your terminal is not already correctly configured, and you desire to do this manually, the settings are as follows...

#### **MMS Application Settings**

- Server: <http://mmsc/>

#### **(WAP) Proxy Settings**

- IP-address: 10.10.10.11
- Port 9201 for WAP stack, or 8080 for TCP/IP.

#### **GPRS Data Settings**

- APN: mms
- Username: Std (Actually, any text will do. Authentication is not based on password)
- Password: Std (Actually, any text will do. Authentication is not based on password)

### ***Test MCPA***

The test instance is used for functional testing of your application, and testing before you receive an account on the production MMSC.

The test MCPA server uses the test MMSC for sending and receiving messages, and you must therefore configure your terminal to use the test MMSC. While your terminal is configured to use the test MMSC, you cannot send messages to subscribers on the production MMSC, or receive messages from subscribers on the production MMSC.

Use the following settings...

#### **MMS Application Settings**

- Server: <http://mmsc:10021/mmsc/>

#### **(WAP) Proxy Settings**

- IP-address: 212.17.144.72

- Port 9201 for WAP stack, or 8080 for TCP/IP stack

### **GPRS Data Settings**

- APN: mmstest
- Username: Std (Actually, any text will do. Authentication is not based on password)
- Password: Std (Actually, any text will do. Authentication is not based on password)

Only registered subscribers may send messages through a MMSC. The test MMSC has a feature called ‘auto-subscription’. You activate your subscription by sending a MMS (to yourself or someone else on the test MMSC). Your sending will fail the first time, and auto-subscription is then activated. Your second sending attempt will succeed, provided that everything works as expected.

## Appendix D – SOAP Implementation Guide

This document primarily describes how to connect to the MCPA server using a Java API. If you do not want to use this API, e.g. because you are using another programming environment than Java, then this appendix may assist you in implementing your own MM7/SOAP client.

MCPA release 2 implements the MM7 interface specified in the MMS Functional Specification [ETSI123v5]. Please note that document version 5.7.0 has been used. MCPA states compliance with MM7 version 5.5.0.

### Implemented MM7 Messages

The following MM7 messages are implemented. For every message, the optional information elements in the specification are listed, with a note on how MCPA implements this. Please refer to the MMS Functional Specification for a list of mandatory and conditional information elements.

#### MM7\_submit.REQ

The table below states the compliance level for the information elements.

Information Element	Presence	MCPA Compliance
Transaction ID	Mandatory	
Message type	Mandatory	
MM7 version	Mandatory	
VASP ID	Optional	Mandatory in MCPA.
VAS ID	Optional	Not used.
Sender address	Optional	Mandatory in MCPA. VAS Provider's short number must be enclosed in <Number>-tags, not <ShortCode>-tags.
Service code	Optional	Mandatory in MCPA. Must contain the billing information string: Pricecategory#ServiceId#ServiceCategory, e.g. 'CPAMMS1000#NEWS#ANYTEXT'
Linked ID	Optional	Not used.
Message class	Optional	Not used.
Date and time	Optional	Not used.
Time of expiry	Optional	Not used.
Earliest delivery time	Optional	Not used.
Delivery report	Optional	Optional in MCPA. Default value is 'false'.
Read reply	Optional	Not used.

Reply-Charging	Optional	Not used.
Reply-Deadline	Optional	Not used.
Reply-Charging-Size	Optional	Not used.
Priority	Optional	Not used.
Subject	Optional	<b>Optional in MCPA.</b>
Adaptations	Optional	Not used.
<b>Charged Party</b>	Optional	<b>Mandatory in MCPA.</b>
<b>Content type</b>	<b>Mandatory</b>	
<b>Content</b>	Optional	<b>Mandatory in MCPA.</b>
Message Distribution Indicator	Optional	Not used.

### MM7\_submit.RES

The table below states the compliance level for the information elements.

Information Element	Presence	MCPA Compliance
<b>Transaction ID</b>	<b>Mandatory</b>	
<b>Message type</b>	<b>Mandatory</b>	
<b>MM7 version</b>	<b>Mandatory</b>	
<b>Message ID</b>	<b>Conditional</b>	
<b>Request Status</b>	<b>Mandatory</b>	
Request Status text	Optional	Not used.

### MM7\_deliver.REQ

The table below states the compliance level for the information elements.

Information Element	Presence	MCPA Compliance
<b>Transaction ID</b>	<b>Mandatory</b>	
<b>Message type</b>	<b>Mandatory</b>	
<b>MM7 version</b>	<b>Mandatory</b>	
MMS Relay/Server ID	Optional	Not used.
Linked ID	Optional	Not used.
<b>Sender address</b>	<b>Mandatory</b>	
<b>Recipient address</b>	Optional	<b>Mandatory in MCPA.</b>
Date and time	Optional	Supported when forwarded from MMSC.

Reply-Charging-ID	Optional	Not used.
Priority	Optional	Not used.
Subject	Optional	Supported.
<b>Content type</b>	<b>Mandatory</b>	
<b>Content</b>	Optional	<b>Mandatory in MCPA.</b>

### MM7\_deliver.RES

The table below states the compliance level for the information elements.

Information Element	Presence	MCPA Compliance
<b>Transaction ID</b>	<b>Mandatory</b>	
<b>Message type</b>	<b>Mandatory</b>	
<b>MM7 version</b>	<b>Mandatory</b>	
Service code	Optional	Not used.
<b>Request Status</b>	<b>Mandatory</b>	
Request Status text	Optional	Not used.

### MM7\_delivery\_report.REQ

The table below states the compliance level for the information elements.

Information Element	Presence	MCPA Compliance
<b>Transaction ID</b>	<b>Mandatory</b>	
<b>Message Type</b>	<b>Mandatory</b>	
<b>MM7 Version</b>	<b>Mandatory</b>	
MMS Relay/Server ID	Optional	Not used.
<b>Message ID</b>	<b>Mandatory</b>	
<b>Recipient address</b>	<b>Mandatory</b>	
<b>Sender address</b>	<b>Mandatory</b>	
<b>Date and time</b>	<b>Mandatory</b>	
<b>MM Status</b>	<b>Mandatory</b>	<ul style="list-style-type: none"> <li>• Retrieved</li> <li>• Rejected</li> <li>• Expired</li> <li>• Forwarded</li> <li>• Indeterminate</li> </ul>
MM Status text	Optional	Not used.

Please refer to the API Javadoc [MCPAAPI] for class `com.telenor.mcpaclient.DeliveryStatus` for details on delivery report statuses.

## MM7\_delivery\_report.RES

The table below states the compliance level for the information elements.

Information Element	Presence	MCPA Compliance
Transaction ID	Mandatory	
Message Type	Mandatory	
MM7 Version	Mandatory	
Request Status	Mandatory	
Request Status text	Optional	Not used.

## Unsupported MM7 Messages

The following messages are not supported by MCPA:

- MM7\_cancel.REQ
- MM7\_cancel.RES
- MM7\_replace.REQ
- MM7\_replace.RES

## Transfer-Encoded Response

If your client submits a request using HTTP 1.1, then MCPA may respond with a chunked encoded response. Support for chunked transfer-encoding is mandatory in HTTP 1.1.

## Connection Testing

You should make sure that your application responds to HTTP GET requests, as described in the section “Implementing HTTP GET for Connection Testing”.