

<b>ID</b>		<b>Creation date</b>	May 5, 2009
<b>Platform</b>	S60 2nd Edition, S60 3rd Edition, S60 5th Edition	<b>Tested on devices</b>	Nokia N95, Nokia E90
<b>Category</b>	M	<b>Subcategory</b>	Messaging

**Keywords (APIs, classes, methods, functions):** sms, mms, mail

## Overview

This article shows how to handle SMS, MMS and e-mail operations in [m](#).

## Preconditions

**Note:** The `mail` module is not available for S60 2nd Edition.

**Note:** The `send` function from the `sms` module requires the `CostComm` permission; the `send` function from the `mms` requires the `CostComm` and the `Read` permissions; the `send` function from the `mail` module requires the `CostComm` and the `ReadApp` permissions.

## SMS

### Sending an SMS

```
use sms

sms.send("1234567890", "This is the message")
```

### Waiting for an SMS and displaying it

```
use sms, time

//Wait for a new SMS to arrive and when it does, store its ID
id = sms.receive()
//Collect information about it
msg = sms.get(id)

//Display the SMS content, sender and time of arrival
content = msg["text"]
sender = msg["sender"]
time = time.str(msg["time"])
```

## Messaging\_in\_m

```
print content + "\n"
print "From: " + sender + "\n"
print "Received at: " + time
```

## Deleting all the read messages and setting the unread ones as read

```
use sms

for id in sms.inbox() do
  if sms.get(id)["unread"] = false then sms.delete(id)
  else sms.set(id, ["unread":false])
end
end
```

## MMS

### Sending an MMS

```
use mms

mms.send("1234567890", "Subject", ["C:\\attachment1.jpg", "C:\\attachment2.mp3"])
```

### Waiting for an MMS and saving its attached files

```
use mms, io

//Wait for a new MMS to arrive and when it does, store its id
id = mms.receive()
//Retrieve the message
msg = mms.get(id)

//Save its attachments
for i = 0 to len(msg["files"]) - 1 do
  file_name = msg["files"][i];
  file_name = substr(file_name, rindex(file_name, "\\") + 1);
  //Create a stream object from the file
  j = mms.open(id, i);
  //Create an empty file to write to using its stream object
  o = io.create("C:\\\" + file_name);
  //Read 256 characters from the attached file's stream
  b = io.read(j, 256);
  while b # null do
    //Write the data to the file we created
    io.write(o, b);
    //Read the next 256 characters from the attached file
    b = io.read(j, 256);
  end;
  //Close the streams
  io.close(j);
end;
```

```
        io.close(0);  
end
```

## E-mail

### Sending an email

```
use mail
```

```
mail.send(["to":"address1@domain1.com", "cc":"address2@domain2.com"], "Subject", "This is the e-m
```

## Postconditions

The operations described above are performed.