

Solbergagymnasiet, Arvika  
Projektarbete 100p  
2004-05-24

# MMS-Avkodare skriven i PHP

Jonatan Heyman  
[jonatan@heyman.info](mailto:jonatan@heyman.info)  
<http://heyman.info/>

## Innehållsförteckning

1. Inledning.....	3
1.1 Bakgrund.....	3
1.2 Syfte/Målsättning.....	3
1.3 Varför öppen källkod?.....	4
2. MMS.....	4
2.1 Teknisk förklaring.....	4
2.2 Möjligheter/Begränsningar.....	5
2.3 Uppbyggnaden av ett MMS-meddelande.....	5
2.3.1 Header-kodning.....	6
2.3.2 Body-kodning.....	8
2.3.3 Exempel på en MMS-PDU.....	9
3. Programmering/Tillvägagångssätt/Svåra moment.....	10
3.1 MMS-avkodare.....	11
3.2 MMS-applikation.....	11
3.3 Svåra moment.....	11
3.4 Publicering.....	12
4. Sammanfattning.....	12
4.2 Användning av exempelapplikationen.....	12
5. Resurser/Informationskällor.....	13
5.1 Specifikationer.....	13
5.2 Web.....	13
6. Bilagor.....	14
6.1 Förkortningslista.....	15
6.2 Loggbok.....	16
6.3 MMS-avkodare <mmsdecoder.php>.....	18
6.4 Webapplikation.....	28
6.4.1 <config.php>.....	29
6.4.2 <dload.php>.....	30
6.4.3 <functions.php>.....	31
6.4.4 <get.php>.....	34
6.4.5 <index.php>.....	35
6.4.6 <list.php>.....	36
6.5 Kopia av Affero General Public License.....	37

# 1. Inledning

MMS (multimedia messaging service) är nya generationen mobiltelefonsnabbmeddelanden och skillnaden mellan MMS och dess föregångare, SMS (smart messaging system), är stora. Förutom att det är en helt ny teknik för att skicka meddelanden, kan man också skicka program, bilder, ljud, film mm. Idag har nästan alla nya telefoner som tillverkas stöd för MMS, vilket givetvis öppnar nya möjligheter för programmerare.

Observera att jag i denna projektavhandling kommer att använda mig av ett visst mått av "svengelska" då många IT-ord inte har någon bra, svensk översättning. Detta är alltså medvetet och beror inte på bristande svenskakunskap.

## 1.1 Bakgrund

Någong gång under hösten 2003 kom jag för första gången (om man bortser från artiklar lästa i tidningar och internet) i kontakt med MMS meddelanden. En kompis visade mig en sida på internet som hade ett system för att i realtid lägga upp bilder som man skickade med MMS. Jag blev mycket imponerad och tog reda på mer information om hur meddelanden skickades. Då jag insåg att uppgiften visserligen skulle vara en utmaning, men inte omöjlig, bestämde jag mig för att göra ett försök. Efter mycket googlande och ögnande i specifikationer kunde jag äntligen börja. Det var efter att jag lyckats avkoda en liten del av headern (ca 100-150 rader kod var då skrivet), som jag seriöst började fundera på att ta detta som mitt projektarbete. Därför började jag skriva ner alla ändringar och förbättringar jag gjorde, i en s.k. changelog\*, för att senare kunna skriva loggbok med hjälp av denna.

*\*Changelog är ett textdokument där man väldigt kortfattat skriver ner vilken förbättring eller ändring som gjorts i programmet. Med hjälp av den kunde jag senare se vad jag hade gjort och när, för att då skriva en utförligare förklaring i loggboken.*

## 1.2 Syfte/Målsättning

Målsättningen jag satte upp var att utveckla ett klassbibliotek i PHP, med funktioner för att ta emot och avkoda MMS-meddelanden, och sedan konstruera en applikation som implementerade biblioteket på ett användbart sätt.

Min applikation skulle fungera som MMSC och ta emot meddelanden utan att sedan skicka en notifiering till mottagarklienten, som en normal MMSC skulle ha gjort.

Jag ville att mitt MMS-bibliotek, om möjligt, skulle kunna avkoda hela headern i ett MMS-meddelande samt ta ut alla delar ur MMS-meddelandet (se 2.3 Uppbyggnaden av ett MMS-meddelande).

Jag ville att webapplikationen som utnyttjade MMS-biblioteket skulle vara enkel och inte allt för

avancerad, eftersom folk som vill bygga egna applikationer som använder sig av mitt MMS-bibliotek enkelt skulle kunna kolla i min exempel-applikation. (se 6.4 Webapplikation).

Koden ska släppas som öppen källkod (se 1.3 Varför öppen källkod?).

### 1.3 Varför öppen källkod?

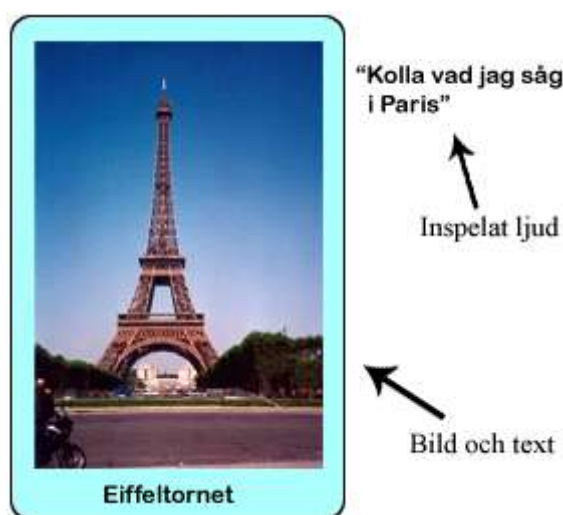
Det finns väldigt många anledningar till varför man bör välja öppen källkod framför stängd, men i detta avsnitt hade jag inte tänkt skriva en uppsats om varför öppen källkod är bättre än stängd, utan helt enkelt lista de huvudsakliga anledningarna att jag valde att publicera detta projekt som öppen källkod.

Jag gör inte detta projekt för någon slags ekonomisk vinning utan enbart för utmaningen och intresset för tekniken. Dessutom tror jag inte att det finns några stora pengar i idén, men att det kan finnas visst intresse för ett fritt MMS-bibliotek bland folk som vill hacka ihop egna små projekt.

Dessutom kan andra utvecklare bidra med buggfixar och förbättringar till ett program som är öppen källkod.

## 2. MMS

MMS har tagits fram för att kunna erbjuda stora möjligheter att skicka meddelande innehållande multimediamaterial så som bild, ljud och film. Exempel på ett MMS meddelande kan ses i Figur 1.

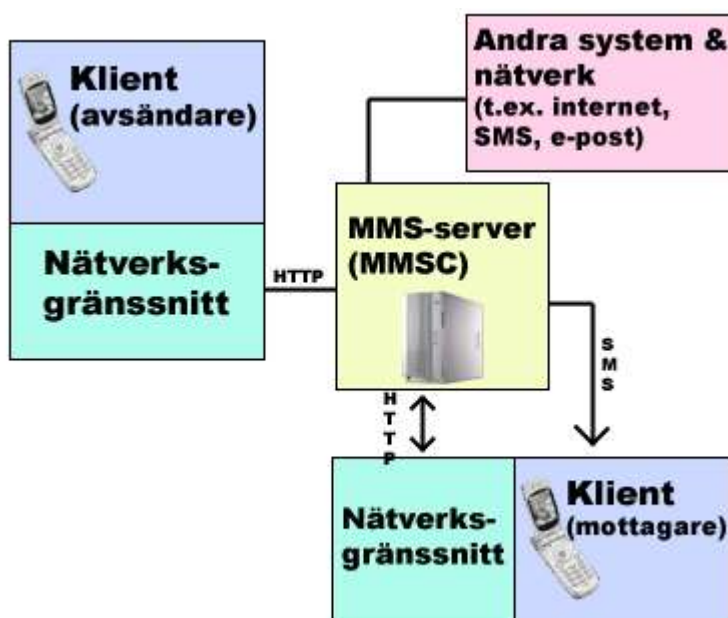


Figur 1. Exempel på ett MMS meddelande

MMS är ett ej-real-tids system för att skicka meddelanden. Det kan jämföras med liknande system, som t.ex. e-mail och SMS och det är tänkt att MMS också ska kunna integreras med sådana befintliga system.

## 2.1 Teknisk förklaring

Då någon skickar ett MMS-meddelande är det första som händer att hans/hennes mobil (om vi förutsätter att de skickar det från en mobiltelefon vilket är vanligast) kodar meddelandet (se 2.3 Uppbyggnaden av ett MMS-meddelande). Därefter ansluter telefonen till operatörens WAP nät. Då telefonen är uppkopplad skickas en HTTP-förfrågan till en MMSC som tar emot hela meddelandet genom HTTP metoden POST. MMSCn lägger då upp meddelandet så att det blir tillgängligt på en http-adress och skickar sedan iväg ett SMS meddelande till mottagaren. Detta SMS är binärkodat och innehåller information om var MMS-meddelandet kan hämtas. Mottagarens telefon meddelar då sin ägare om att det finns ett MMS-meddelande tillgängligt och han/hon kan då välja att telefonen ska koppla upp sig via GPRS/WAP, hämta hem meddelandet och sedan visa det. I figur två kan man se hur ett förenklat MMS-nätverk kan se ut.



Figur 2. Exempel på hur ett MMS-nätverk kan se ut

## 2.2 Möjligheter/Begränsningar

Möjligheten att integrera med andra system är en av MMS-systemets nyckelfunktioner. Förutom att kunna skicka multimediameddelanden till sina vänner, är det också möjligt att med hjälp av MMS skicka e-post och få väderrapporter.

Mitt projekt utökar ytterligare möjligheterna med MMS, eftersom det ger vem som helst med en webserver, möjligheten att starta egna MMS-tjänster. Användningsområden för mitt projekt som jag har kommit på, förutom att kunna lägga upp bilder på websidor i realtid, är bland annat att kunna utföra kommandon på servern (starta om, eller kanske starta en grann-dator med hjälp av "Wake on LAN"), skicka billiga SMS (via någon gratis internet-SMS-tjänst) och utföra saker i online spel

(t.ex. Utopia <http://games.swirve.com/utopia/>).

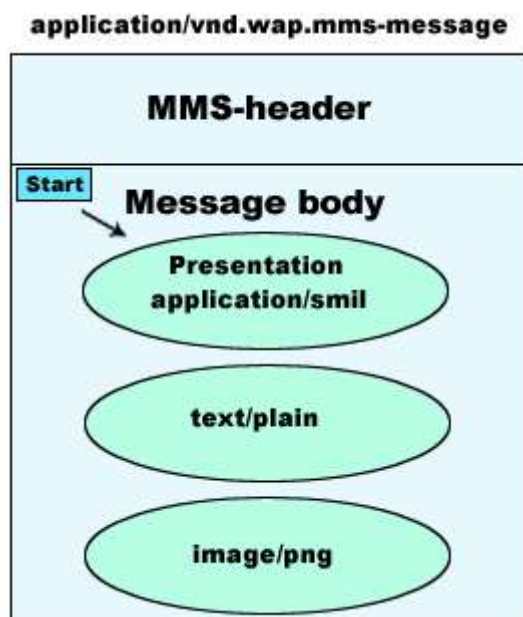
De begränsningar som finns sätts ofta av telefonbolagen. En begränsning som i stort sett alltid finns är kostnaden. Telia har kostnaden ungefär 1.96 öre per kilobyte på sitt GPRS nät, vilket ändå är mycket lite om man jämför med vad det kostar att skicka ett MMS via deras MMSC (2.90 kr / MMS). Vissa telefoner kan också vara begränsade av telefonbolagen, så att de enbart kan skicka MMS-meddelanden via deras MMSC.

## 2.3 Uppbyggnaden av ett MMS-meddelande

Som tidigare nämnts, kodas ett MMS-meddelande innan det skickas, för att det ska ta så liten plats som möjligt. För att meddelandet sedan ska kunna läsas, måste det avkodas, vilket också är själva kärnan i mitt projektarbete.

Det är inte meningen att jag ska skriva om hela MMS-kodnings-specifikationen här, och jag tänker därför fatta mig ganska kortfattat. För vidare information om hur kodningen går till hänvisar jag till "wap-209-mmsencapsulation" och "WAP-230-WSP" specifikationerna, som ges ut av Open Mobile Alliance (<http://www.openmobilealliance.org/>), samt koden i bilaga 5.6.1.

Det finns flera olika MMS PDUer ("protocol data unit"), men de har alla många likheter med varandra. De PDUer jag kommer ta upp är "*m-send-req*", som används då ett MMS skickas till en MMSC, och "*m-send-conf*", som används då MMSCn svarar på ett m-send-req. En "*m-send-req*"-PDU består av en MMS-header och en MMS-body. Headern innehåller information om MMS-meddelandet, som t.ex. avsändare, mottagare, prioritet, ämne etc. Body-delen har en content-type, som vanligtvis är "*application/vnd.wap.multipart.related*" eller "*application/vnd.wap.multipart.mixed*" och innehåller i dessa fall multimediaobjekt som också kallas "parts", eller "delar" på svenska. En av delarna kan vara en presentationsdel (vanligtvis av typen "*application/smil*") och i dessa fall pekar MMS-meddelandets content-type-parameter "start" på denna del. Presentationsdelen talar om för klienten hur denna skall presentera MMS-meddelandets innehåll.



Figur 3. Så här kan ett ganska typiskt MMS-meddelande vara uppbyggt.

Body-delen skickats enbart då ett meddelande skickas och hämtas. Alla andra PDUer skickar bara header-delen.

### 2.3.1 Header-kodning

Headern i ett MMS-meddelande är uppbyggt på följande vis. Först en ensam byte med den mest signifikanta biten satt till 1, som indikerar vilket header-fält som därefter kommer. Beroende på vilket fält det är, representeras värdet i fältet olika. I Tabell 1, som är hämtad från ”wap-209-mmsencapsulation” specifikationen med vissa tillägg från min sida, visas vilka header-fält som existerar för PDU-typen ”m-send-req”.

Tabell 1, lista över m-send-req header-fält

<i>Name</i>	<i>Content</i>	<i>Egen kommentar</i>
X-Mms-Message-Type	Message-type-value = m-send-req	Denna representeras aen ensam byte, i detta fall <octet 128>.
X-Mms-Transaction-ID	Transaction-id-value	Unikt id för varje MMS-meddelande. Represesenters av en text-string.

<i>Name</i>	<i>Content</i>	<i>Egen kommentar</i>
X-Mms-MMS-Version	MMS-version-value	Versionsnumret representeras av en ensam byte med den mest signifikanta biten satt till 1, där det första versionsnumret (major) finns i de tre mest signifikanta bitarna efter den första, och det andra versionsnumret (minor) finns i de återstående 4 bitarna.
Date	Date-value	Datomet representeras av en Long-integer och är ett unix-timestamp. Ett unix-timestamp indikerar tiden genom antalet sekunder sedan 1970-01-01 00:00:00.
From	From-value	Innehåller avsändarens telefonnummer/adress och är kodat som en Encoded-string-value, som förutom en vanlig textsträng också kan innehålla charset. Se specifikationen för hur detta kodas.
To	To-value	Minst en av fälten To, CC och BCC måste finnas med i ett MMS. Kodas som Encoded-string-value.
Cc	Cc-value	Värdet representeras av Encoded-string-value.
Bcc	Bcc-value	Värdet representeras av Encoded-string-value.
Subject	Subject-value	Värdet representeras av Encoded-string-value.
X-Mms-Message-Class	Message-class-value	Detta värde avkodas, om den mest signifikanta biten i byten efter fältindikatorn är satt, som en ensam byte där 128 = personal, 129 = advertisement, 130 = informational, 131 = auto.  Annars kodas message-class-value som en textstring.
X-Mms-Expiry	Expiry-value	Kodas likadant som Date-value.
X-Mms-Delivery-Time	Delivery-time-value	
X-Mms-Priority	Priority-value	Representeras av en ensam byte där 128 = låg, 129 = normal och 130 = hög.
X-Mms-Sender-Visibility	Sender-visibility-value	Representeras av en ensam byte som boolean, där 128 är sant och 129 falskt.
X-Mms-Delivery-Report	Delivery-report.value	Kodas likadant som Sender-visibility-value.
X-Mms-Read-Reply	Read-reply-value	Representeras av en ensam byte som boolean. 128 = sant och 129 = falskt.
Content-type	Content-type-value	Content-type fältet är utan tvekan det allra svåraste att avkoda. Avkodningen av detta fält beskrivs senare i denna avhandling.

Då ”*m-send-req*” har skickats, svarar MMSCn med ett ”*m-send-conf*”. I Tabell 2 nedan, kan man se hur en sådan PDU är uppbyggd.

<i>Name</i>	<i>Content</i>	<i>Egen kommentar</i>
X-Mms-Message-Type	Message-type-value = m-send-conf	Reprenteras av en ensam byte, i detta fall <octet 129>.
X-Mms-Transaction-ID	Transaction-id-value	Detta är samma transaktions id, som användes i ”m-send-req” förfrågan. Representeras av en text-string.
X-Mms-MMS-Version	MMS-version-value	Se Tabell 1.
X-Mms-Response-Status	Response-status-value	Response status. 128 = Ok
X-Mms-Response-Text	Response-text-value	Ej nödvändig. Text som matchar reponse-status
Message-ID	Message-ID-value	Ej nödvändig. Används för att ge rapport då meddelanden har mottagits.

### 2.3.2 Body-kodning

MMS-bodyn är uppbyggd som multipart data, som beskrivs i kapitel 8.5 i WAP-230-WSP specifikationen. Multipart data består av en header och ett antal poster. Se Figur 4.



Figur 4. Application/vnd.wap.multipart Formatet.

Headern består i sin tur av ett enda fält vilket är nEntries, som talar om hur många poster/delar som finns i bodyn.

Därefter följer delarna direkt efter varandra, och de är kodade på följande vis:

<i>Namn</i>	<i>Typ</i>	<i>Syfte</i>
HeadersLen	Uintvar	Längden på postens headersfält och dess ContentType.
DataLen	Uintvar	Längden på postens datafält.
ContentTye	Multiple octets	Postens Content-type.
Headers	HeaderLen – (length of ContentType) antal bytes lång.	Postens header.
Data	DataLen antal bytes stor.	Innehåller postens data.

### 2.3.3 Exempel på en MMS-PDU

I Figur 5 syns ett exempel på en header tillhörande ett MMS-meddelandes.

```

8C 80 98 34 2D 66 63 36|30 00 8D 90 85 04 40 B0 | 0€~4-fc60 00...J@°
BF A4 89 01 81 97 31 32|33 34 35 2F 54 59 50 45 | ç¼%r[-12345/TYPE
3D 50 4C 4D 4E 00 96 48|65 6A 00 8A 80 8F 81 94 | =PLMN -Hej Š€00~
81 86 81 90 81 84 1B B3|89 61 70 70 6C 69 63 61 | 0+000,←³%applica
74 69 6F 6E 2F 73 6D 69|6C 00 8A 3C 41 41 41 41 | tion/smil Š<AAAA
3E 00 03 22 12 0C 83 85|6D 6D 73 2E 74 78 74 00 | > L"↓f...mms.txt
    
```

Figur 5. Headerfälten är uppdelade i vartannat blått, vartannat rött område, med den första byten som dess indikator.

Nedan följer en förklaring till fälten i MMS-meddelandets header.

Hex	Förklaring
8C	Message-type
80	m-send-req
98	Transaction-id
34 2D 66 63 36 30	Bla bla
00	End of string
8D	Mms-version
90	0x90 = 10010000 Major=001=1 Minor=0000=0 Version=1.0
85	Date
04	Värdet är sparat i 4 bytes.
40 B0 BF A4	$(40 \ll 24) + (B0 \ll 16) + (BF \ll 8) + A4 = 1085325220$ s. sedan 1970-01-01 00:00:00
89 01 81	From = ""
97	To
31 32 33 34 35 3F 54 59 50 45 3D 50 4C 4D 4E	12345/TYPE=PLMN
00	End of string
96	Subject
48 65 6A	Hej
00	End of string

<i>Hex</i>	<i>Förklaring</i>
8A	Message-class
80	128 = Personal
8F	Priority
81	129 = Normal
94	Sender-visibility
81	No
86	Delivery-report
81	No
90	Read-reply
81	No
84	Content-type
1B	Content-type är sparad som content-general-form Längden är 27
B3	Content-type representeras av en ensam byte med den mest signifikanta biten satt till 1. Content-type = 0x33 = application/vnd.wap.multipart.related
89	Content-type-parameter: Start
61 70 70 6C 69 63 61 74 69 6F 6E 2F 73 6D 69 6C	application/smil
00	End of string
8A	Content-type-parameter: Type
3C 41 41 41 41 3E	Representerad som en text-string. "<AAAA>"
00	End of string
...	Vi är nu inne i "application/vnd.wap.multipart.

### 3. Programmering/Tillvägagångssätt/Svåra moment

I denna del av avhandlingen tar jag upp hur jag har gått tillväga, hur jag tänkt och vad jag har haft problem med.

Det första jag gjorde då jag startade med projektet var att jag började kolla runt på nätet efter information om hur MMS-tekniken fungerar. Informationen som fanns, var ganska knapp, men jag lyckades så småningom leta mig fram till de rätta specifikationerna. Det var ganska tydligt att specifikationerna var skrivna av och för folk som var vana att jobba med "parsers" och specifikationer, vilket försvårade för mig, då detta var något ganska nytt för min del. Efter att ha

ögnat igenom ”wap-209-mmsencapsulation”-specifikationen ett otal gånger, kunde jag trots allt börja sätta ihop någon slags prototyp till en parser.

### 3.1 MMS-avkodare

Att skriva avkodaren som en PHP-klass, var för mig en självklarhet, eftersom jag ville att den skulle vara lätt att implementera i en applikation.

Jag valde att spara MMS-datan i en array, med en byte per plats, istället för att ha den i en sträng, eftersom jag ansåg att parsningen skulle bli lättare då. Jag hade en variabel som indikerade vilken byte som parsern för tillfället befann sig på. Sedan skrev jag ihop en funktion som läste in vilket headerfält det var vid den aktuella positionen och sedan läste in fältets värde. Denna kördes i en while-loop som avslutades då funktionen returnerade false. Funktionen returnerade false då det inte fanns mer data (något hade blivit fel), eller då content-type fältet blivit parsat, eftersom headern då var slut. Detta var själva grund ”motorn” i parsern, och jag byggde sedan på fler funktioner allt eftersom.

När headerparsningen var i stort sett klar, konstruerade jag MMSPart klassen, vars uppgift var att hålla ordning på de olika posterna/delarna i ett MMS. I dess konstruktor sparar den lite variabler som den får tillskickat sig. Funktionen save() skrevs för att kunna spara part-datan i en fil på servern.

### 3.2 MMS-applikation

När MMS-avkodaren var skriven, var en stor del av projektet klart. För att ha ett exempel som implementerade MMS-avkodaren skrev jag därför en enkel applikation. Enkelheten är avsiktlig då det ska vara lätt för andra att med hjälp av exempel-applikationen skriva egna, mer avancerade och mer användbara applikationer.

Jag började med att skriva en funktion som sparade ett mottaget MMS-meddelande i databasen. Därefter gjorde jag en funktion som kunde lista MMS-meddelandena och PHP-script som använde sig av funktionerna. Applikationen svarade till en början inte med ett ”m-send-conf” svar, vilket ledde till att klienten trodde att meddelandet inte kommit fram, då därför implementerades även en funktion för att svara med en giltigt ”m-send-req”-sträng.

För att applikationen skulle bli lättare att använda, skapade jag också en install-fil, vars uppgift var att skapa tabeller i databasen efter att man ställt in rätt inställningar i config-filen.

### 3.3 Svåra moment

Under arbetet dök det givetvis upp svåra moment när det gäller avkodningen. Många berodde på oklarheter i specifikationerna och andra på att en del saker var nya för mig. De flesta har dock kunnat lösas med hjälp av ”ord-för-ord”-studier av specifikationerna och andra har krävt att jag frågat folk jag känner, så väl som folk på olika forum, om hjälp. Det finns dock några som utmärker sig mer än andra och dessa tänkte jag ta upp här.

Avkodningen av MMS-meddelandets content-type var utan tvekan det svåraste momentet jag stötte på under hela projektet. Det var svårt att hitta bra information om hur den skulle avkodas, och specifikationen var svårförstådd. Nu när jag väl vet hur kodningen fungerar, och går tillbaka till specifikationen, tycker jag dock inte att den är svårförståelig. För info om hur ett MMS-meddelandes content-type avkodas, hänvisar jag till specifikationen och den medföljande PHP-koden.

Jag hade också problem med hur jag skulle avkoda värden av typen encoded-string-value och faktum är att MMS-avkodaren fortfarande inte stöder olika charsets. Som tur är är detta inte något som påverkar avkodarens funktionalitet fatalt, men är givetvis något som ska fixas.

Även skickandet av "m-send-conf"-svaret bjöd på problem och det var väldigt svårt att debugga då den enda output telefonen (SonyEricsson T310) gav, var att det blev fel. Det visade sig efter mycket om och men att hela felet berodde på att jag glömt sätta HTTP-headerns content-type.

### 3.4 Publicering

När jag till slut var klar med programmeringen, var det dags för att förbereda scripten för publicering. Jag städade ut gamla kodsuttag som inte användes längre och skrev ihop en liten readme-fil. Filer som inte skulle vara med i releasen rensades undan och allt packades ihop till två stycken filer. En som enbart innehöll mmsdecoder-klassen och en som innehöll både den och exempelapplikationen.

En enkel hemsida sattes ihop och scripten publicerades.

<http://heyman.info/mmsdecoder/>

## 4. Sammanfattning

Sammanfattningsvis kan jag säga att detta projektarbete har varit både roligt och mycket lärorikt. Det har många gånger bjudit på utmaningar, och jag måste säga att jag vissa gånger trodde jag tagit mig vatten över huvud.

Jag är nöjd med resultatet, vilket är en nästan fullständig MMS-avkodare, och jag planerar att utveckla den vidare tills den blir fullständig, samt en exempel-applikation som använder sig av MMS-avkodaren.

Jag ser många möjligheter med klassen, och jag planerar redan nu på att implementera den i en ny funktion till min och några kompisars "nätmötesplats" (<http://www.helvetet.org>). Tanken är att användarna ska kunna skicka upp bild och text till sidan, med hjälp av MMS. T.ex. ska bilderna hamna i olika gallerier (personligt eller allmänt), beroende på vilket mottagarnummer de skriver in.

Det är min förhoppning att någon ska ladda ner koden, få användning för någon del av den, och kanske t.o.m. bidra med egen kod.

Den som vill se applikationen ”in action” kan gå in på:

<http://helvetet.org/jonatan/mmsapp/>

Och för att skicka upp MMS:

<http://helvetet.org/jonatan/mmsapp/get.php>

## 4.2 Användning av exempelapplikationen

Att installera exempelapplikationen är lätt för den som har viss vana att arbeta med PHP-script, men här kommer en kort beskrivning. Beskrivningen förutsätter att man har en fungerande webserver med PHP och MySQL som körs.

1. Ladda upp filerna i ”mmsdecoder\_application.zip” till en katalog på servern.
2. Gör inställningar i config.php.
3. Kör install.php.
4. Ta bort eller döp om install.php

Det kan vara lite klurigare att ställa in sin MMS-telefon att skicka MMS till sitt script. Detta är givetvis individuellt för varje telefonmodell, men om man letar efter en inställning som heter något i stil med ”MMSC” eller ”MMS-server” bör man kunna hitta det.

Jag har tyvärr inte haft tillgång till så många MMS-telefoner under arbetet med detta projektarbete, så jag kan därför inte inkludera instruktioner för olika telefonmodeller.

## 5. Resurser/Informationskällor

Under arbetets gång har jag givetvis tagit hjälp av olika informationskällor. I denna avdelning tänkte jag försöka lista så många av dessa som möjligt.

### 5.1 Specifikationer

Jag har använt mig av följande specifikationer:

wap-209-mmsencapsulation-20020105-a  
WAP-230-WSP-20010705-a  
wap-205-mmsarchoverview-20010425-a

### 5.2 Web

Följande websidor har varit till stor hjälp:

<http://www.nowsms.com>

<http://www.openmobilealliance.org/>

<http://www.forum.nokia.com/main.html>

## **6. Bilagor**

## 6.1 Förkortningslista

<i>Förkortning</i>	<i>Förklaring</i>
MMS	Multimedia Messaging Service
SMS	Short Messaging Service
PDU	Protocol Data Unit
WAP	Wireless Application Protocol
HTTP	Hyper Text Transfer Protocol

## 6.2 Loggbok

Denna loggbok innehåller ändringar jag gjort i MMS-avkodarens källkoden. Tidpunkter då jag inhämtat kunskap från specifikationer, arbete med exempel applikationen och arbete med rapporten finns alltså inte med. Changeloggen påbörjades också efter att delar av koden redan var färdigskriven.

### 2004-03-02

Prototypen för parsern är byggd. Jag skapade en changelog för att senare kunna skriva en loggbok.

Skapade prototypen för en debugfunktion som skulle kunna skriva ut HEX-kod, för att underlätta vid debugandet.

### 2004-03-15

Lade till fler kommentarer i koden, som dokumenterade hur headeravkodningen fungerade.

Parsern kollar nu om "message-type" är "m-send-req" och slutar i annat fall parse. Parsern slutar också parse hedern efter att "content-type" har blivit parsat, eftersom det alltid är det sista fältet i headern.

Skapade en funktion som skulle parse variabeltypen "content-general-form".

### 2004-03-16

Lade till kod för att ta ut expiry-fältet

Ändrade i filen rawdata.php, som sparar all data som finns i \$HTTP\_RAW\_POST\_DATA i en fil, så att det även sparades en info-fil, med information från HTTP-headern, som t.ex. USER\_AGENT.

### 2004-03-17

Fixade några buggar.

Testade avkodaren med flera PDU-filer som jag hittade på nätet, med positivt resultat.

### 2004-03-18

Började skriva ihop kod för att avkoda ett MMS-meddelandes "content-type"-fält.

### 2004-03-21

Bestämde mig för att släppa projektet under Affero GPL, pga. den sk. ASP-loophole, som existerar i GPL.

Skrev ihop en prototyp för "MMS Part"-klassen.

#### **2004-04-15**

Skrev ihop kod till HEX-debug funktionen som jag tidigare bara skrivit skalet för. Funktionen fungerar nu korrekt.

Fixade en bugg i parseUint() funktionen. Positionsvariabeln blev inte ökad om den mest signifikanta biten inte var 1.

#### **2004-05-10**

Fixade parseUint() funktionen som fortfarande var buggig. Det tog mig lång tid och jag fick krångla rejält, men kommer förhoppningsvis att funka fint.

Fixade bättre stöd för avkodningen av "content-type"-fältet

Lade till funktionen parseParts() som körs i en for-loop, efter att headern har blivit avkodad. Lade till lite kod för att ta ut data ur delarna för att se om jag kunde få ut någon data som verkade vettig. Det visade sig att datan togs ut korrekt, då "content-type"-fältet parsades korrekt, vilket det gjorde i två av mina PDU-filer.

#### **2004-05-12**

Lade till bättre stöd för stöd för att ta ut content-typen. Eftersom jag totalt ignorerade content-type-parametrarna var det inte så konstigt att det inte fungerade. De två PDUer som fungerade råkade inte ha några parametrar. Efter denna fixen verkade alla PDUer avkodas korrekt! Låt oss nu avkoda partheadern så är vi snart klara för släppning!

Gaaah! "Content-type"-fältet tar kol på mig! När jag lade till samma avkodningsmekanism för mms-delarna, funkade bara avkodningen i vissa fall. Så jag antar att något är fel med avkodningen. Jag får väl fortsätta debugga. :(

Fortsatte att greja med "content-type"-avkodningen, men inget fixades. Jag känner doack att jag är nära en lösning :).

#### **2004-05-13**

Lade till nästan fullt stöd för avkodningen av "content-type"-fältet. Avkodningen av parametervärdena bör förbättras och "content-type"-avkodningen för MMSet och delarna är fortfarande separerade på två ställen, vilket givetvis borde bli fixat.

#### **2004-05-15**

Header och body delen är nu parsad fint. Testade med en ericsson-telefon och det fungerade bra.

**2004-05-23**

Rensade upp i koden och den första publika versionen (0.75) är redo att släppas!

### 6.3 MMS-avkodare <mmsdecoder.php>

```

<?php
// Version 0.75 //

/**
 * Copyright (C) 2004 Jonatan Heyman
 *
 * This file is part of PHP MMS Decoder.
 *
 * PHP MMS library is free software; you can redistribute it and/or
 * modify it under the terms of the Affero General Public License as
 * published by Affero, Inc.; either version 1 of the License, or
 * (at your option) any later version.
 *
 * PHP MMS library is distributed in the hope that it will be
 * useful, but WITHOUT ANY WARRANTY; without even the implied warranty
 * of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * Affero General Public License for more details.
 *
 * You should have received a copy of the Affero General Public
 * License in the COPYING file that comes with The Affero Project; if
 * not, write to Affero, Inc., 510 Third Street, Suite 225, San
 * Francisco, CA 94107 USA.
 */

define( "DEBUG", 1 ); // Print parseerrors? Print values while they are parsed?

/*-----*
 * Constants *
 *
 * http://wapforum.org/ *
 * WAP-209-MMSEncapsulation-20020105-a *
 * Table 8 *
 *
 * The values are encoded using WSP 7bit encoding. *
 * Read more about how to decode this here: *
 * http://www.nowSMS.com/discus/messages/12/3287.html *
 *
 * Example from the above adress: *
 * 7Bit 0D = 0001101 *
 * 8Bit 0D = 10001101 = 8D *
 *-----*/
define( "BCC", 0x81 );
define( "CC", 0x82 );
define( "CONTENT_LOCATION", 0x83 );
define( "CONTENT_TYPE", 0x84 );
define( "DATE", 0x85 );
define( "DELIVERY_REPORT", 0x86 );
define( "DELIVERY_TIME", 0x87 );
define( "EXPIRY", 0x88 );
define( "FROM", 0x89 );
define( "MESSAGE_CLASS", 0x8A );
define( "MESSAGE_ID", 0x8B );
define( "MESSAGE_TYPE", 0x8C );
define( "MMS_VERSION", 0x8D );
define( "MESSAGE_SIZE", 0x8E );
define( "PRIORITY", 0x8F );
define( "READ_REPLY", 0x90 );
define( "REPORT_ALLOWED", 0x91 );
define( "RESPONSE_STATUS", 0x92 );
define( "RESPONSE_TEXT", 0x93 );
define( "SENDER_VISIBILITY", 0x94 );
define( "STATUS", 0x95 );
define( "SUBJECT", 0x96 );
define( "TO", 0x97 );
define( "TRANSACTION_ID", 0x98 );

/*-----*
 * Array of header contents *
 *-----*/
$mmsMessageTypes = array (
    0x80 => "m-send-req",
    0x81 => "m-send-conf",
    0x82 => "m-notification-ind",
    0x83 => "m-notifyresp-ind",

```

```

    0x84 => "m-retrieve-conf",
    0x85 => "m-acknowledge-ind",
    0x86 => "m-delivery-ind",
    0x00 => NULL
);

/*-----*
 * Some other useful arrays *
 *-----*/
$mmsYesNo = array(
    0x80 => 1,
    0x81 => 0,
    0x00 => NULL
);

$mmsPriority = array(
    0x80 => "Low",
    0x81 => "Normal",
    0x82 => "High",
    0x00 => NULL
);

$mmsMessageClass = array(
    0x80 => "Personal",
    0x81 => "Advertisement",
    0x82 => "Informational",
    0x83 => "Auto"
);

$mmsContentTypes = array(
    0x00 => '*/*',
    0x01 => 'text/*',
    0x02 => 'text/html',
    0x03 => 'text/plain',
    0x04 => 'text/x-hdml',
    0x05 => 'text/x-ttml',
    0x06 => 'text/x-vCalendar',
    0x07 => 'text/x-vCard',
    0x08 => 'text/vnd.wap.wml',
    0x09 => 'text/vnd.wap.wmlscript',
    0x0A => 'text/vnd.wap.wta-event',
    0x0B => 'multipart/*',
    0x0C => 'multipart/mixed',
    0x0D => 'multipart/form-data',
    0x0E => 'multipart/byteranges',
    0x0F => 'multipart/alternative',
    0x10 => 'application/*',
    0x11 => 'application/java-vm',
    0x12 => 'application/x-www-form-urlencoded',
    0x13 => 'application/x-hdmlc',
    0x14 => 'application/vnd.wap.wmlc',
    0x15 => 'application/vnd.wap.wmlscriptc',
    0x16 => 'application/vnd.wap.wta-eventc',
    0x17 => 'application/vnd.wap.uaprof',
    0x18 => 'application/vnd.wap.wtls-ca-certificate',
    0x19 => 'application/vnd.wap.wtls-user-certificate',
    0x1A => 'application/x-x509-ca-cert',
    0x1B => 'application/x-x509-user-cert',
    0x1C => 'image/*',
    0x1D => 'image/gif',
    0x1E => 'image/jpeg',
    0x1F => 'image/tiff',
    0x20 => 'image/png',
    0x21 => 'image/vnd.wap.wbmp',
    0x22 => 'application/vnd.wap.multipart.*',
    0x23 => 'application/vnd.wap.multipart.mixed',
    0x24 => 'application/vnd.wap.multipart.form-data',
    0x25 => 'application/vnd.wap.multipartbyteranges',
    0x26 => 'application/vnd.wap.multipart.alternative',
    0x27 => 'application/xml',
    0x28 => 'text/xml',
    0x29 => 'application/vnd.wap.wbxml',
    0x2A => 'application/x-x968-cross-cert',
    0x2B => 'application/x-x968-ca-cert',
    0x2C => 'application/x-x968-user-cert',
    0x2D => 'text/vnd.wap.si',
    0x2E => 'application/vnd.wap.sic',
    0x2F => 'text/vnd.wap.sl',
    0x30 => 'application/vnd.wap.slc',

```

```

0x31 => 'text/vnd.wap.co',
0x32 => 'application/vnd.wap.coc',
0x33 => 'application/vnd.wap.multipart.related',
0x34 => 'application/vnd.wap.sia',
0x35 => 'text/vnd.wap.connectivity-xml',
0x36 => 'application/vnd.wap.connectivity-wbxml',
0x37 => 'application/pkcs7-mime',
0x38 => 'application/vnd.wap.hashed-certificate',
0x39 => 'application/vnd.wap.signed-certificate',
0x3A => 'application/vnd.wap.cert-response',
0x3B => 'application/xhtml+xml',
0x3C => 'application/wml+xml',
0x3D => 'text/css',
0x3E => 'application/vnd.wap.mms-message',
0x3F => 'application/vnd.wap.rollover-certificate',
0x40 => 'application/vnd.wap.locc+wbxml',
0x41 => 'application/vnd.wap.loc+xml',
0x42 => 'application/vnd.syncml.dm+wbxml',
0x43 => 'application/vnd.syncml.dm+xml',
0x44 => 'application/vnd.syncml.notification',
0x45 => 'application/vnd.wap.xhtml+xml',
0x46 => 'application/vnd.wv.csp.cir',
0x47 => 'application/vnd.oma.dd+xml',
0x48 => 'application/vnd.oma.drm.message',
0x49 => 'application/vnd.oma.drm.content',
0x4A => 'application/vnd.oma.drm.rights+xml',
0x4B => 'application/vnd.oma.drm.rights+wbxml'
);

/*-----*
 * The MMS header decoding class *
 *-----*/
class MMSDecoder {
    var $data; // The unparsed MMS data in an array of the ascii numbers
    var $pos = 0; // The current parsing position of the data array
    var $PARTS = array();

    // The parsed data will be saved in these variables
    var
        $BCC,
        $CC,
        $CONTENTLOCATION,
        $CONTENTTYPE,
        $DATE,
        $DELIVERYREPORT,
        $DELIVERYTIME,
        $EXPIRY,
        $FROM,
        $MESSAGECLASS,
        $MESSAGEID,
        $MESSAGETYPE,
        $MMSVERSIONMAJOR,
        $MMSVERSIONMINOR,
        $MESSAGELENGTH,
        $PRIORITY,
        $READREPLY,
        $REPORTALLOWED,
        $RESPONSESTATUS,
        $RESPONSETEXT,
        $SENDERVISIBILITY,
        $STATUS,
        $SUBJECT,
        $TO,
        $TRANSACTIONID,
        $MMSVERSIONRAW, // used for the m-send-conf (confirmation answer)
        $CONTENTTYPE_PARAMS; // parameter-values for the MMS content-type

    // Constructor
    function MMSDecode($data) {
        $this->data = array();

        // Save the data in an array containing the ascii numbers
        for ($i = 0; $i < strlen($data); $i++)
            $this->data[$i] = ord($data[$i]);

        // Reset position

```

```

        $this->pos = 0;

        // Reset variables
        $this->PARTS = array();
    }

    // This function is called when the data is to be parsed
    function parse() {
        // Reset position
        $this->pos = 0;

        // parse the header
        while ($this->parseHeader());

        // Header done, fetch parts, but make sure the header was parsed correctly
        if ($this->CONTENTTYPE == 'application/vnd.wap.multipart.related' || $this->CONTENTTYPE ==
'application/vnd.wap.multipart.mixed')
            while ($this->parseParts());
        else
            return 0;

        return 1;
    }

    /*-----*
    * This function checks what kind of field is to be *
    * parsed at the moment *
    * *
    * If true is returned, the class will go on and *
    * and continue decode the header. If false, the *
    * class will end the header decoding. *
    *-----*/
    function parseHeader() {
        // Some global variables used
        global $mmsMessageTypes, $mmsYesNo, $mmsPriority, $mmsMessageClass, $mmsContentTypes;

        if (!array_key_exists($this->pos, $this->data))
            return 0;

        switch ($this->data[$this->pos++]) {
            case BCC:
                $this->BCC = $this->parseEncodedStringValue();
                if (DEBUG) $this->debug("BCC", $this->BCC);
                break;
            case CC:
                $this->CC = $this->parseEncodedStringValue();
                if (DEBUG) $this->debug("CC", $this->CC);
                break;
            case CONTENT_LOCATION:
                $this->CONTENTLOCATION = $this->parseTextString();
                if (DEBUG) $this->debug("Content-location", $this->CONTENTLOCATION);
                break;
            case CONTENT_TYPE:
                if ($this->data[$this->pos] <= 31) { /* Content-general-form */
                    $len = $this->parseValueLength();

                    // check if next byte is in range of 32-127. Then we have a Extension-media
                    which is a textstring
                    if ($this->data[$this->pos] > 31 && $this->data[$this->pos] < 128)
                        $this->CONTENTTYPE = $this->parseTextString();
                    else {
                        // we have Well-known-media; which is an integer
                        $this->CONTENTTYPE = $mmsContentTypes[$this->parseIntegerValue()];
                    }
                } elseif ($this->data[$this->pos] < 128) { /* Constrained-media - Extension-media*/
                    $this->CONTENTTYPE = $this->parseTextString();
                } else /* Constrained-media - Short Integer*/
                    $this->CONTENTTYPE = $mmsContentTypes[$this->parseShortInteger()];

                // Ok, now we have parsed the content-type of the message, let's see if there are
                any parameters
                while (!$noparams) {
                    switch ($this->data[$this->pos]) {
                        case 0x89: // Start, textstring
                            $this->pos++;
                            $this->parseTextString();
                            break;

```

```

        case 0x8A: // type, constrained media
            $this->pos++;
            if ($this->data[$this->pos] < 128) { /* Constrained-media - Extension-
media*/
                $this->pos++;
                $this->parseTextString();
            } else // Constraiind-media Short Integer
                $this->CONTENTTYPE_PARAMS[type] = $this->parseShortInteger();
            break;
        default:
            $noparams = 1;
            break;
    }
}

if (DEBUG) $this->debug("Content-type", $this->CONTENTTYPE);

// content-type parsed, that means we have reached the end of the header
return 0;

case DATE: /* In seconds from 1970-01-01 00:00 GMT */
    $this->DATE = $this->parseLongInteger();
    if (DEBUG) $this->debug("Date", date("Y-m-d H:i:s", $this->DATE));
    break;
case DELIVERY_REPORT: /* Yes | No */
    $this->DELIVERYREPORT = $mmsYesNo[ $this->data[$this->pos++] ];
    if (DEBUG) $this->debug("Delivery-report", $this->DELIVERYREPORT);
    break;
case DELIVERY_TIME:
    if (DEBUG) $this->debug("Delivery-time", $this->DELIVERYTIME);
    break;
case EXPIRY:
    // not sure if this is right, but if I remeber right, it's the same format as
date...
    $this->EXPIRY = parseLongInteger();
    if (DEBUG) $this->debug("Expiry", $this->EXPIRY);
    break;
case FROM:
    $this->FROM = $this->parseEncodedStringValue();
    if (DEBUG) $this->debug("From", $this->FROM);
    break;
case MESSAGE_CLASS:
    $this->MESSAGECLASS = $mmsMessageClass[ $this->parseMessageClassValue() ];
    if (DEBUG) $this->debug("Message-class", $this->MESSAGECLASS);
    break;
case MESSAGE_ID: /* Text string */
    $this->MESSAGEID = $this->parseTextString();
    if (DEBUG) $this->debug("Message-id", $this->MESSAGEID);
    break;
case MESSAGE_TYPE:
    //$this->MESSAGETYPE = $mmsMessageTypes[ $this->data[$this->pos++] ];
    $this->MESSAGETYPE = $this->data[$this->pos++];

    // check that the message type is m-send-req
    if (!$this->MESSAGETYPE = 128)
        debug("Wrong type", "The message-type field is not 'm-send-req' (Octet 128)",
1);

    if (DEBUG) $this->debug("Message-type", $mmsMessageTypes[$this->MESSAGETYPE]);
    break;
case MMS_VERSION:
    /*
    * The version number (1.0) is encoded as a WSP short integer, which
    * is a 7 bit value.
    *
    * The three most significant bits (001) are used to encode a major
    * version number in the range 1-7. The four least significant
    * bits (0000) contain a minor version number in the range 1-14.
    */
    $this->MMSVERSIONRAW = $this->data[$this->pos];
    $this->MMSVERSIONMAJOR = ($this->data[$this->pos] & 0x70) >> 4;
    $this->MMSVERSIONMINOR = ($this->data[$this->pos++] & 0x0F);

    if (DEBUG) $this->debug("MMS-version", $this->MMSVERSIONMAJOR . "." . $this-
>MMSVERSIONMINOR);
    break;
case MESSAGE_SIZE: /* Long integer */
    $this->MESSAGESIZE = $this->parseLongInteger();
    if (DEBUG) $this->debug("Message-size", $this->MESSAGESIZE);

```

```

        break;
    case PRIORITY:          /* Low | Normal | High */
        $this->PRIORITY = $mmsPriority[ $this->data[ $this->pos++ ] ];
        if (DEBUG) $this->debug("Priority", $this->PRIORITY);
        break;
    case READ_REPLY:       /* Yes | No */
        $this->READREPLY = $mmsYesNo[ $this->data[ $this->pos++ ] ];
        if (DEBUG) $this->debug("Read-reply", $this->READREPLY);
        break;
    case REPORT_ALLOWED:   /* Yes | No */
        $this->REPORTALLOWED = $mmsYesNo[ $this->data[ $this->pos++ ] ];
        if (DEBUG) $this->debug("Report-allowed", $this->REPORTALLOWED);
        break;
    case RESPONSE_STATUS:
        $this->RESPONSESTATUS = $this->data[ $this->pos++ ];
        if (DEBUG) $this->debug("Response-status", $this->RESPONSESTATUS);
        break;
    case RESPONSE_TEXT:    /* Encoded string value */
        $this->RESPONSETEXT = $this->parseEncodedStringValue();
        if (DEBUG) $this->debug("Response-text", $this->RESPONSETEXT);
        break;
    case SENDER_VISIBILITY: /* Hide | show */
        $this->SENDERVISIBILITY = $mmsYesNo[ $this->data[ $this->pos++ ] ];
        if (DEBUG) $this->debug("Sender-visibility", $this->SENDERVISIBILITY);
        break;
    case STATUS:
        $this->STATUS = $this->data[ $this->pos++ ];
        if (DEBUG) $this->debug("Status", $this->STATUS);
        break;
    case SUBJECT:
        $this->SUBJECT = $this->parseEncodedStringValue();
        if (DEBUG) $this->debug("Subject", $this->SUBJECT);
        break;
    case TO:
        $this->TO = $this->parseEncodedStringValue();
        if (DEBUG) $this->debug("To", $this->TO);
        break;
    case TRANSACTION_ID:
        $this->TRANSACTIONID = $this->parseTextString();
        if (DEBUG) $this->debug("Transaction-id", $this->TRANSACTIONID);
        break;
    default:
        if ($this->data[ $this->pos - 1 ] > 127)
            $this->debug("Parse error", "Unknown field (" . $this->data[ $pos-1 ] . ")!",
                $this->pos-1);
        else
            debug("Parse error:", "Value encountered when expecting field!", $this->pos);
        break;
    }

    return true;
}

/*-----*
 * Function called after header has been parsed. This function fetches *
 * the different parts in the MMS. Returns true until it encounter end *
 * of data. *
 *-----*/
function parseParts() {
    global $mmsContentTypes; // for parsing the contenttypes

    if (!array_key_exists($this->pos, $this->data))
        return 0;

    // get number of parts
    $count = $this->parseUint();

    if (DEBUG) $this->debug("MMS parts", $count);

    for ($i = 0; $i < $count; $i++) {
        // new part, so clear the old data and header
        $data = "";
        $header = "";
        unset($ctype);

        // get header and data length
        $headerlen = $this->parseUint();
        $datalen = $this->parseUint();

```

```

/* PARSE CONTENT-TYPE */
// this is actually the same structure as in the MMS content-type
// so maybe we should make this in a better way, but for now, I'll
// just cut n paste

// right now I just save the position in the MMS data array before I parse
// the content-type, to be able to roll back after it has been parsed because
// the headerlen includes both the content-type and the header
// TODO: this is just a fast hack and should be done in a more proper way
$typepos = $this->pos;

if ($this->data[$this->pos] <= 31) { /* Content-general-form */
    // the value follows after the current byte and is "current byte" long
    $len = $this->parseValueLength();

    // check if next byte is in range of 32-127. Then we have a Extension-media which is
a textstring
    if ($this->data[$this->pos] > 31 && $this->data[$this->pos] < 128)
        $ctype = $this->parseTextString();
    else {
        // we have Well-known-media; which is an integer
        $ctype = $mmsContentTypes[$this->parseIntegerValue()];
    }
} elseif ($this->data[$this->pos] < 128) { /* Constrained-media - Extension-media*/
    //$this->pos++;
    $ctype = $this->parseTextString();
} else /* Constrained-media - Short Integer */
    $ctype = $mmsContentTypes[$this->parseShortInteger()];

// roll back position so it's just before the content-type again
$this->pos = $typepos;
/* END OF CONTENT TYPE */

// Read header. Actually, we don't do anything with this yet.. just skipping it (note
that the content-type is included in the header)
for ($j = 0; $j < $headerlen; $j++)
    $header .= chr($this->data[$this->pos++]);

// read data
for ($j = 0; $j < $datalen; $j++)
    $data .= chr($this->data[$this->pos++]);

if (DEBUG) $this->debug("Part ($i):headerlen", $headerlen);
if (DEBUG) $this->debug("Part ($i):datalen", $datalen);
if (DEBUG) $this->debug("Part ($i):content-type", $ctype);
//if (DEBUG) $this->debug("Part ($i):data", $data); // I've commented this one, to get a
cleaner debug

    $this->PARTS[] = new MMSPart($headerlen, $datalen, $ctype, $header, $data);
}

return false;
}

/*-----*
* Parse message-class *
* message-class-value = Class-identifier | Token-text *
* Class-identifier = Personal | Advertisement | Informational | Auto *
*-----*/
function parseMessageClassValue() {
    if ($this->data[$this->pos] > 127) {
        // the byte is one of these 128=personal, 129=advertisement, 130=informational, 131=auto

        return $this->data[$this->pos++];
    } else
        return $this->parseTextString();
}

/*-----*
* Parse Text-string *
* text-string = [Quote <Octet 127>] text [End-string <Octet 00>] *
*-----*/
function parseTextString() {
    // Remove quote
    if ($this->data[$this->pos] == 0x7F)
        $this->pos++;
}

```

```

        while ($this->data[$this->pos])
            $str .= chr($this->data[$this->pos++]);

        $this->pos++;

        return $str;
    }

    /*-----*
    * Parse Encoded-string-value *
    * *
    * Encoded-string-value = Text-string | Value-length Char-set Text-string *
    * *
    *-----*/
function parseEncodedStringValue() {
    if ($this->data[$this->pos] < 32) {
        $len = $this->parseValueLength();

        $this->pos++;

        for ($i = 0; $i < $len-1; $i++)
            $str .= chr( $this->data[$this->pos++] );
        return $str;
    } else
        return $this->parseTextString();
}

/*-----*
* Parse Value-length *
* Value-length = Short-length<Octet 0-30> | Length-quote<Octet 31> Length<Uint> *
* *
* A list of content-types of a MMS message can be found here: *
* http://www.wapforum.org/wina/wsp-content-type.htm *
*-----*/
function parseValueLength() {
    if ($this->data[$this->pos] < 31) {
        // it's a short-length
        return $this->data[$this->pos++];
    } elseif ($this->data[$this->pos] == 31) {
        // got the quote, length is an Uint
        $this->pos++;
        return $this->parseUint();
    } else {
        // uh, oh... houston, we got a problem
        die("Parse error: Short-length-octet (" . $this->data[$this->pos] . ") > 31 in Value-
length at offset " . $this->pos . "!\n");
    }
}

/*-----*
* Parse Long-integer *
* Long-integer = Short-length<Octet 0-30> Multi-octet-integer<1*30 Octets> *
*-----*/
function parseLongInteger() {
    // Get the number of octets which the long-integer is stored in
    $octetcount = $this->data[$this->pos++];

    // Error checking
    if ($octetcount > 30)
        die("Parse error: Short-length-octet (" . $this->data[$this->pos-1] . ") > 30 in Long-
integer at offset " . $this->pos-1 . "!\n");

    // Get the long-integer
    for ($i = 0; $i < $octetcount; $i++) {
        $longint = $longint << 8;
        $longint += $this->data[$this->pos++];
    }

    return $longint;
}

/*-----*
* Parse Short-integer *
* Short-integer = OCTET *
* Integers in range 0-127 shall be encoded as a one octet value with the *

```

```

    * most significant bit set to one, and the value in the remaining 7 bits *
    *-----*/
function parseShortInteger() {
    return $this->data[$this->pos++] & 0x7F;
}

/*-----*
 * Parse Integer-value *
 * Integer-value = short-integer | long-integer *
 * *
 * This function checks the value of the current byte and then *
 * calls either parseLongInt() or parseShortInt() depending on *
 * what value the current byte has *
 *-----*/
function parseIntegerValue() {
    if ($this->data[$this->pos] < 31)
        return $this->parseLongInteger();
    elseif ($this->data[$this->pos] > 127)
        return $this->parseShortInteger();
    else {
        $this->debug('ERROR', 'Not a IntegerValue field', $this->pos);
        $this->pos++;
        return 0;
    }
}

/*-----*
 * Parse Unsigned-integer *
 * *
 * The value is stored in the 7 last bits. If the first bit is set, *
 * then the value continues into the next byte. *
 * *
 * http://www.nowsms.com/discus/messages/12/522.html *
 *-----*/
function parseInt() {
    //if (!$this->data[$this->pos] & 0x80)
    //    return $this->data[$this->pos++] & 0x7F;

    while ($this->data[$this->pos] & 0x80) {
        // Shift the current value 7 steps
        $uint = $uint << 7;
        // Remove the first bit of the byte and add it to the current value
        $uint |= $this->data[$this->pos++] & 0x7F;
    }

    // Shift the current value 7 steps
    $uint = $uint << 7;
    // Remove the first bit of the byte and add it to the current value
    $uint |= $this->data[$this->pos++] & 0x7F;

    return $uint;
}

/*-----*
 * Function which outputs debug messages *
 *-----*/
function debug($name, $str, $pos = -1, $errorlevel = 0) {
    if ($pos != -1)
        echo "<b>$name ($pos):</b> " . $str;
    else
        echo "<b>$name:</b> " . $str;

    echo "<br>\n";

    if ($errorlevel > 0)
        exit;
}

/*-----*
 * Function to output a part of the mmsdata *
 * in HEX form, and mark one byte with a ^ *
 *-----*/
function debughex($start, $count, $markstart = -1, $markcount = -1) {
    $hexcount = 0;
    $markstop = $markstart + $markcount;

```

```

// set font so that the hex will be more readable
echo '<br><font face="fixedsys" size="-1">';

// loop thru data and print hex
for ($i = $start; $i <= ($start+$count); $i++) {
    // fix marking
    if ($i == $markstart)
        echo '<font color="#ff0000">';
    if ($i == $markstop)
        echo '</font>';

    $hex = dechex($this->data[$i]);

    // add 0 before hex if needed
    if (strlen($hex) < 2)
        $hex = '0' . $hex;
    // add space
    $hex = ' ' . $hex;

    // check hexcount, wrap lines if needed etc
    if ($hexcount == 8)
        echo '| ';
    elseif ($hexcount == 16) {
        echo '<br>';
        $hexcount = 0;
    }
    $hexcount++;

    echo $hex;
}

// som more html
echo '</font><br><br>';
}
}

/*-----*
* The MMS part class *
* An instance of this class contains the one parts of an MMS message. *
* *
* The multipart type is formed as: *
* number |part1|part2|...|partN *
* where part# is formed by headerlen|datalen|contenttype|headers|data *
*-----*/
class MMSPart {
    var $headerlen;
    var $header;
    var $DATALEN;
    var $CONTENTTYPE;
    var $DATA;

    /*-----*
    * Constructor, just store the data *
    *-----*/
    function MMSPart($headerlen, $datalen, $ctype, $header, $data) {
        $this->hpos = 0;
        $this->headerlen = $headerlen;
        $this->DATALEN = $datalen;
        $this->CONTENTTYPE = $ctype;
        $this->DATA = $data;
    }

    /*-----*
    * Save the data to a location on disk *
    *-----*/
    function save($filename) {
        $fp = fopen($filename, 'wb');
        fwrite($fp, $this->DATA);
        fclose($fp);
    }
}
?>

```

## **6.4 Webapplikation**

### 6.4.1 <config.php>

```
<?php

/**
 * Copyright (C) 2004 Jonatan Heyman
 *
 * This file is part of PHP MMS Decoder Example Application.
 * Configuration file
 *
 * PHP MMS library is free software; you can redistribute it and/or
 * modify it under the terms of the Affero General Public License as
 * published by Affero, Inc.; either version 1 of the License, or
 * (at your option) any later version.
 *
 * PHP MMS library is distributed in the hope that it will be
 * useful, but WITHOUT ANY WARRANTY; without even the implied warranty
 * of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * Affero General Public License for more details.
 *
 * You should have received a copy of the Affero General Public
 * License in the COPYING file that comes with The Affero Project; if
 * not, write to Affero, Inc., 510 Third Street, Suite 225, San
 * Francisco, CA 94107 USA.
 */

// MySQL settings
define('MYSQL_HOST', 'localhost');
define('MYSQL_USER', ''); // please note that it is a security risk to run and PHP scripts as
'root'
define('MYSQL_PASS', '');
define('MYSQL_DB', 'mms');

// Database table settings
define('TABLE_MMS', 'mms');
define('TABLE_PARTS', 'parts');

define('SAVE_RAWDATA', 0); // if the raw post data shall be saved in files

?>
```

## 6.4.2 <dload.php>

```
<?php
/**
 * Copyright (C) 2004 Jonatan Heyman
 *
 * This file is part of PHP MMS Decoder Example Application.
 * Download/show a part of an MMS stored in the database.
 *
 * PHP MMS library is free software; you can redistribute it and/or
 * modify it under the terms of the Affero General Public License as
 * published by Affero, Inc.; either version 1 of the License, or
 * (at your option) any later version.
 *
 * PHP MMS library is distributed in the hope that it will be
 * useful, but WITHOUT ANY WARRANTY; without even the implied warranty
 * of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * Affero General Public License for more details.
 *
 * You should have received a copy of the Affero General Public
 * License in the COPYING file that comes with The Affero Project; if
 * not, write to Affero, Inc., 510 Third Street, Suite 225, San
 * Francisco, CA 94107 USA.
 */

require_once('config.php');
require_once('functions.php');

db_connect();

mms_part_dload($_GET[id]);

db_close();

?>
```

### 6.4.3 <functions.php>

```

<?php
/**
 * Copyright (C) 2004 Jonatan Heyman
 *
 * This file is part of PHP MMS Decoder Example Application.
 * A collection of functions used by the application.
 *
 * PHP MMS library is free software; you can redistribute it and/or
 * modify it under the terms of the Affero General Public License as
 * published by Affero, Inc.; either version 1 of the License, or
 * (at your option) any later version.
 *
 * PHP MMS library is distributed in the hope that it will be
 * useful, but WITHOUT ANY WARRANTY; without even the implied warranty
 * of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * Affero General Public License for more details.
 *
 * You should have received a copy of the Affero General Public
 * License in the COPYING file that comes with The Affero Project; if
 * not, write to Affero, Inc., 510 Third Street, Suite 225, San
 * Francisco, CA 94107 USA.
 */

/**
 * It is safe to assume that the config.php file has already been included here
 */

// Save an MMS and its parts to the database
// Takes only a reference variable to a MMSDecode object as argument
function mms_save(&$mms) {
    // save MMS in database
    $sql = "
        INSERT INTO " . TABLE_MMS . "
        (
            `from`,
            `to`,
            `subject`,
            `content_type`
        ) VALUES (
            '" . mysql_escape_string($mms->FROM) . "',
            '" . mysql_escape_string($mms->TO) . "',
            '" . mysql_escape_string($mms->SUBJECT) . "',
            '" . mysql_escape_string($mms->CONTENTTYPE) . "'
        )
    ";
    //echo $sql;

    if (!mysql_query($sql))
        log_error('Database error: Could not insert MMS data into database. MySQL: ' . mysql_error());

    $mmsid = mysql_insert_id();

    // loop thru parts and save them in database
    foreach ($mms->PARTS as $part) {
        $result = mysql_query("INSERT INTO " . TABLE_PARTS . " (mmsid, datalen, content_type, data)
VALUES ('$mmsid', '" . $part->DATALEN . "', '" . $part->CONTENTTYPE . "', '" . mysql_escape_string($part->DATA) . "')");
        if (!$result)
            log_error('Database error: Could not insert part data into database. MySQL: ' . mysql_error());
    }
}

// Get the MMS and their parts from the database and generate some HTML
function mms_list($ret = 0) {
    $sql = "
        SELECT
            " . TABLE_MMS . ".from,
            " . TABLE_MMS . ".to,
            " . TABLE_MMS . ".subject,
            " . TABLE_MMS . ".content_type as mms_content_type,
            " . TABLE_PARTS . ".id,
            " . TABLE_PARTS . ".mmsid,
    
```

```

        " . TABLE_PARTS . ".datalen,
        " . TABLE_PARTS . ".content_type
FROM
    " . TABLE_MMS . ", " . TABLE_PARTS . "
WHERE
    " . TABLE_PARTS . ".mmsid = " . TABLE_MMS . ".id
ORDER BY
    " . TABLE_PARTS . ".mmsid DESC
";

$result = mysql_query($sql);

$lastid = -1;
$html = '<table border="1">';

while ($rsdata = mysql_fetch_assoc($result)) {
    if ($lastid != $rsdata[mmsid]) {
        $html .= '
            <tr><td>
                Subject: ' . $rsdata[subject] . '<br>
                From: ' . $rsdata[from] . '<br>
                To: ' . $rsdata[to] . '<br>
                Content-type: ' . $rsdata[mms_content_type] . '
            <br><br>
        ';
    }

    $html .= '<a href="dload.php?id=' . $rsdata[id] . '">' . $rsdata[content_type] . '</a> ' .
        $rsdata[datalen] . ' bytes<br>';

    // check if part-content-type is png, jpeg, tiff or gif.. then output picture
    if ($rsdata[content_type] == 'image/gif' || $rsdata[content_type] == 'image/jpeg' || $rsdata
[content_type] == 'image/png' || $rsdata[content_type] == 'image/tiff')
        $html .= '<br><br>';

    $lastid = $rsdata[mmsid];

    if ($lastid != $rsdata[mmsid])
        $html .= '</td></tr>';
}

$html .= '</td></tr></table>';

if (!$ret)
    echo $html;
else
    return $html;
}

function mms_part_dload($id) {
    $result = mysql_query("SELECT content_type, data FROM " . TABLE_PARTS . " WHERE id='$id'");

    if (!$result)
        echo 'Part not found!';
    else {
        $rsdata = mysql_fetch_assoc($result);
        header('Content-type: ' . $rsdata[content_type]);
        header('Content-Disposition: inline; filename="part"');
        echo $rsdata[data];
    }
}

// Send an OK response to the sender after the MMS has been recieved
// See "6.1.2. Send confirmation" in the wap-209-mmsencapsulation specification, on how this is
constructed
function mms_confirm (&$mms) {
    $pos = 0;

    $confirm[$pos++] = 0x8C; // message-type
    $confirm[$pos++] = 129; // m-send-conf
    $confirm[$pos++] = 0x98; // transaction-id

    for ($i = 0; $i < strlen($mms->TRANSACTIONID); $i++)
        $confirm[$i+$pos] = ord(substr($mms->TRANSACTIONID, $i, 1));

    $pos += $i;

    $confirm[$pos++] = 0x00; // end of string

```

```
$confirm[$pos++] = 0x8D; // version
$confirm[$pos++] = 0x90; // 1.0
$confirm[$pos++] = 0x92; // response-status
$confirm[$pos] = 128; // OK

// respond with the m-send-conf
foreach ($confirm as $byte)
    echo chr($byte);
}

// log stuff so it doesn't need to be written out
function log_error($str) {
    echo $str . '<br>';
}

// connect to database
function db_connect() {
    // connect
    if (!mysql_connect(MYSQL_HOST, MYSQL_USER, MYSQL_PASS)) {
        log('Could not connect to database!');
        return 0;
    }

    // select database
    if (!mysql_select_db(MYSQL_DB)) {
        log('Database ' . MYSQL_DB . ' not found!');
        return 0;
    }

    return 1;
}

// close database link
function db_close() {
    mysql_close();
}

?>
```

#### 6.4.4 <get.php>

```

<?php

/**
 * Copyright (C) 2004 Jonatan Heyman
 *
 * This file is part of PHP MMS Decoder Example Application.
 * Recieve an MMS from the client.
 *
 * PHP MMS library is free software; you can redistribute it and/or
 * modify it under the terms of the Affero General Public License as
 * published by Affero, Inc.; either version 1 of the License, or
 * (at your option) any later version.
 *
 * PHP MMS library is distributed in the hope that it will be
 * useful, but WITHOUT ANY WARRANTY; without even the implied warranty
 * of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * Affero General Public License for more details.
 *
 * You should have received a copy of the Affero General Public
 * License in the COPYING file that comes with The Affero Project; if
 * not, write to Affero, Inc., 510 Third Street, Suite 225, San
 * Francisco, CA 94107 USA.
 */

// includes
require_once('mmsdecoder.php');
require_once('config.php');
require_once('functions.php');

if ($HTTP_RAW_POST_DATA != "") {
    // check if the raw post data shall be saved
    if (SAVE_RAWDATA) {
        // save RAW data
        $data = $HTTP_RAW_POST_DATA;
        $filename = md5($data . time() . rand(1, 1000));

        $file = fopen($filename, 'wb');
        fwrite($file, $data);
        fclose($file);

        $info = print_r($_SERVER, true);
        $file = fopen($filename . "_info", 'wb');
        fwrite($file, $info);
        fclose($file);
    }

    // parse MMS
    $mms = new MMSDecoder($HTTP_RAW_POST_DATA);
    $mms->parse();

    // connect to database
    db_connect();

    // save mms and it's parts
    mms_save($mms);

    // close db connection
    db_close();

    // set header
    header('Content-type: application/vnd.wap.mms-message');

    // send confirmation response
    mms_confirm($mms);
}

?>

```

### 6.4.5 <index.php>

```
<?php
// redirect to the list.php file
header('Location: list.php');
?>
```

**6.4.6 <list.php>**

```

<?php
/**
 * Copyright (C) 2004 Jonatan Heyman
 *
 * This file is part of PHP MMS Decoder Example Application.
 * List the MMS and their parts.
 *
 * PHP MMS library is free software; you can redistribute it and/or
 * modify it under the terms of the Affero General Public License as
 * published by Affero, Inc.; either version 1 of the License, or
 * (at your option) any later version.
 *
 * PHP MMS library is distributed in the hope that it will be
 * useful, but WITHOUT ANY WARRANTY; without even the implied warranty
 * of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * Affero General Public License for more details.
 *
 * You should have received a copy of the Affero General Public
 * License in the COPYING file that comes with The Affero Project; if
 * not, write to Affero, Inc., 510 Third Street, Suite 225, San
 * Francisco, CA 94107 USA.
 */

require_once('config.php');
require_once('functions.php');

echo '
<html>
<head>
  <title>MMS list</title>
</head>
<body>
  This is PHP MMS Decoder Example Application running. <br>
  To send an MMS to this site, setup your phone MMSC to http://this.address/get.php and send an MMS
  (the number doesn\'t matter). <br><br>

  To download the sourcecode:<br>
  <li><a href="http://heyman.info/mmsdecoder/mmsdecoder.zip">PHP MMS Decoder</a><br>
  <li><a href="http://heyman.info/mmsdecoder/mmsdecoder_application.zip">PHP MMS Decoder
  Example Application</a><br><br>

  To visit the homepage, <a href="http://heyman.info/mmsdecoder/">click here<a>.<br><br>

  Below is a list of the MMS-messages in the database.
  <hr>
  <br>
';

// generation time
$start = microtime(true);

db_connect();
mms_list();
db_close();

$time = microtime(true) - $start;

echo '
<hr>
  PHP MMS Decoder Example Application 1.0. MMS list generated in ' . $time . ' seconds.
</body>
</html>
';
?>

```

## 6.5 Kopia av Affero General Public License

### AFFERO GENERAL PUBLIC LICENSE

Version 1, January 2002

Copyright © 2002 Affero Inc.  
510 Third Street - Suite 225, San Francisco, CA 94107, USA

This license is a modified version of the GNU General Public License copyright (C) 1989, 1991 Free Software Foundation, Inc. made with their permission. Section 2(d) has been added to cover use of software over a computer network.

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the Affero General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This Public License applies to most of Affero's software and to any other program whose authors commit to using it. (Some other Affero software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. This General Public License is designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

#### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this Affero General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

\* a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

\* b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

\* c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is

no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

\* d) If the Program as you received it is intended to interact with users through a computer network and if, in the version you received, any user interacting with the Program was given the opportunity to request transmission to that user of the Program's complete source code, you must not remove that facility from your modified version of the Program or work based on the Program, and must offer an equivalent opportunity for all users interacting with your Program through a computer network to request immediate transmission by HTTP of the complete source code of your modified version or other derivative work.

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

\* a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

\* b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

\* c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other

circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. Affero Inc. may publish revised and/or new versions of the Affero General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by Affero, Inc. If the Program does not specify a version number of this License, you may choose any version ever published by Affero, Inc.

You may also choose to redistribute modified versions of this program under any version of the Free Software Foundation's GNU General Public License version 3 or higher, so long as that version of the GNU GPL includes terms and conditions substantially equivalent to those of this license.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by Affero, Inc., write to us; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE

PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.