

MarshallSoft Client Mailer Library for C/C++

Programmer Manual

(MCM4C)

Version 3.1

January 11, 2012

*This software is provided as-is.
There are no warranties, expressed or implied.*

Copyright (C) 2012
All rights reserved

MarshallSoft Computing, Inc.
Post Office Box 4543
Huntsville AL 35815

Email: info@marshallsoft.com
Web: www.marshallsoft.com

MARSHALLSOFT is a registered trademark of MarshallSoft Computing.

TABLE OF CONTENTS

1	Introduction	Page 3
1.1	MCM Features	Page 4
1.2	Documentation Set	Page 5
1.3	Filenames	Page 5
1.4	MCM Files	Page 5
1.5	Installation	Page 6
1.6	Uninstalling	Page 6
1.7	Purchase	Page 6
1.8	Adding MCM Functions to an Existing Program	Page 7
1.9	KeyCode and Edition Key	Page 7
2	Supported Compilers	Page 7
2.1	Microsoft Visual C++	Page 7
2.2	Borland C/C++	Page 7
2.3	Borland C++ Builder	Page 7
2.4	Watcom C/C++	Page 7
2.5	LCC-Win32 C	Page 7
2.6	MinGW GCC	Page 7
3	The TestMCM Example Program	Page 8
4	The SendMail Example Program	Page 8
5	The GetReply Example Program	Page 9
6	Compiling Example Programs	Page 10
6.1	Compiling TestMCM	Page 10
6.2	Compiling SendMail	Page 10
6.3	Compiling GetReply	Page 10
7	Revision History	Page 11

1 Introduction

The **MarshallSoft Client Mailer for C/C++ (MCM4C)** provides the capability to send **personalized** email to your clients or customers **directly** from your C/C++ application program. The most current version of the **MarshallSoft Client Mailer for C/C++** can be found at <http://www.marshallsoft.com/mcm4c.htm>.

The “**MarshallSoft Client Mailer for C/C++ Programmer Manual**” contains C/C++ programming specific details such as compilers, compiling and running example programs.

The **MarshallSoft Client Mailer (MCM)** functions can be called from both console mode and GUI mode programs. The three example programs (**TestMCM**, **SendMail** and **GetReply**) are Windows 32-bit console mode programs designed to run from a command window.

The **MarshallSoft Client Mailer** works with all versions of Windows (95/98/ME/2000/2003/NT/XP/Vista/Win7 x64.) The **MarshallSoft Client Mailer DLL (MCM32.DLL)** can also be used from any language (Visual C++, .NET, Visual Basic, VB.NET, ACCESS, EXCEL, VBA, Delphi, Visual FoxPro, dBASE, Xbase, etc) capable of calling the Windows API. A Win32 DLL is provided.

Since all programming language versions (C/C++, Visual Basic, Delphi, FoxPro, dBase, and XBase++) of MCM use the identical DLL (MCM32.DLL), once one programming language version is purchased, the **MarshallSoft Client Mailer** can be used with all supported languages.

Regardless of the programming language used, both the email letter to be sent (text or HTML) and the list of recipients are regular ASCII text files. These letter files can be created with any text editor (or HTML editor for HTML files). The recipient list file can be created with either a text editor, exported from a database (such as Access, FoxPro, dBase, Xbase++) or a spreadsheet (such as Excel), or created by a program such as VBA code in Access.

See Section 4 for details on the **SendMail** program and Section 5 for details on the **GetReply** program.

1.1 MCM Features

The features of the **MarshallSoft Client Mailer (MCM)** include:

- Send personalized email **directly** from an application, programmable database or spreadsheet.
- Uses macro substitution strings to personalize outgoing letters (Also called email merge).
- Can send plain text, rich text, or HTML formatted letters.
- Supports multiple attachments.
- Can handle arbitrarily large lists of recipients.
- Detects duplicate email addresses.
- Supports multiple lists of recipients that should always be skipped.
- Supports ASCII, UTF8, and ISO_8859 character sets.
- Supports WIN_1250, WIN_1252, and WIN_1255 character sets.
- Can use up to 32 connections simultaneously for fast email delivery.
- Can process "undeliverable mail" replies.
- Can process multiple user replies ("REMOVE", "CONFIRM", etc...).
- Has extensive error detection and logging capability.
- Includes **SendMail** example program (with source) to send email.
- Includes **GetReply** example program (with source) to read replies.
- Works with all versions of Microsoft Visual C++ (v4.0 through Visual Studio 2010).
- Works with Borland C/C++ (v5.0, v5.5) and Borland C++ Builder.
- Works with Microsoft Foundation Class, Watcom v11, MinGW, and LCC-WIN32.
- Can be called from any program that is capable of calling the Windows API.
- **MCM** functions can be called directly from both ANSI C programs and from C++ programs.
- License includes one year of technical support and downloadable updates.
- Royalty free distribution with your compiled application.
- The license can be used with all supported computer programming languages.

1.2 Documentation Set

There are five manuals in Adobe PDF format for the **MarshallSoft Client Mailer**.

- **Tutorial Manual:** Introduces the basic functionality of the **MarshallSoft Client Mailer**.
- **Servers Manual :** Covers background information on SMTP & POP3 servers.
- **User Manual:** Covers language independent information (letter & list preparation, etc...).
- **Reference Manual:** Contains details for each individual MCM function.
- **Programmer Manual:** Contains language (C/C++, VB, etc.) specific information.

The manuals can be found in the DOCS subdirectory in the **MarshallSoft Client Mailer** file structure when it is installed.

- **Tutorial Manual:** mcm_tutorial.pdf
- **Servers Manual:** mcm_servers.pdf
- **User Manual:** mcm_users.pdf
- **Reference Manuals**

mcm4c_reference.pdf (C/C++)	mcm4d_reference.pdf (Delphi)
mcm4fp_reference.pdf (FoxPro)	mcm4vb_reference.pdf (Visual Basic)
mcm4db_reference.pdf (dBase)	mcm4xb_reference.pdf (Xbase++)
- **Programmer Manuals**

mcm4c_programmer.pdf (C/C++)	mcm4d_programmer.pdf (Delphi)
mcm4fp_programmer.pdf (FoxPro)	mcm4vb_programmer.pdf (Visual Basic)
mcm4db_programmer.pdf (dBase)	mcm4xb_programmer.pdf (Xbase++)

Links to the above manuals can also be found online on the **MarshallSoft Client Mailer** product page:

<http://www.marshallsoft.com/mcm.htm>.

1.3 Filenames

Filename conventions used are listed below.

- Recipient list filenames begin with "List" and end with ".txt"
- Text letter filenames begin with "Letter" and end with ".txt"
- HTML letter filenames begin with "Letter" and end with ".htm"
- HTML template filenames begin with "Template" and with ".htm"
- Background tile filenames begin with "Tile" and end with ".jpg"
- Image filenames begin with "Image" and end with ".jpg"

1.4 MCM Files

mcm32.dll MCL Dynamic Link Library (DLL) file
mcm32.lib MCM Library (LIB) file
mcm.h MCM C/C++ function prototype & constants file

1.5 Installation

(1) Before installation of **MCM4C**, your Windows C/C++ compiler should already be installed on your system and tested. In particular, include command line tools when installing the compiler if you want to compile using command line makefiles.

(2) Unzip MCM4C31.ZIP using any Windows unzip program.

(3) Run the installation program SETUP.EXE which will install all MCM4C files, including copying MCM32.DLL to the Windows directory. Note that DLL registration is not required.

After SETUP is run, the MCM4C files are copied to the directory specified (default \MCM4C). Four sub-directories are created, as follows:

- DOCS - Documentation files
- APPS - Examples program and files
- DLLS - MCM32.DLL
- SSL - SSL proxy server files

1.6 Uninstalling

Uninstalling MCM4C is very easy. First, delete MCM32.DLL that can be found in the \WINDOWS directory (directory \WINNT for Windows NT/2000).

Second, delete the MCM4C project directory created when MCM4C was installed.

1.7 Purchase

Refer to the MCM Users Manual (`mcm_users.pdf` or online at <http://www.marshallsoft.com/mcm.htm>) for pricing and details on the four versions of MCM (personal, professional, professional-plus, and unlimited).

MCM4C can be ordered at

<http://www.marshallsoft.com/order.htm>.

1.8 Adding MCM Functions To an Existing Program

In order to call **MCM** functions from an existing program, (1) add

```
#include "mcm.h"
```

to your application source code, (2) link with MCM32.LIB (for MSVC), MCM32BCB.LIB (Borland C/C++ and C++ Builder), MCM32.LIB (Watcom), or MCM32LCC (Win32/LCC), and recompile from source.

1.9 KeyCode and Edition Key

When a developer license is purchased, the developer will receive a new MCM32.DLL as well as a unique KeyCode and an Edition key for the MCM DLL. Pass the KeyCode as the first argument to **mcmAttach** and the Edition key as the second argument. The KeyCode will be found in the file named "keycode.h" and the Edition key will be found in the file named "edition.h". The KeyCode and edition key are both 0 for the evaluation version. The KeyCode and Edition key for the registered version will be unique 9 or 10 digit numbers. The Edition Key identifies which version of MCM was purchased. Note: Your KeyCode is NOT your Customer ID/Registration number.

2 Supported Compilers

2.1 Microsoft Visual C/C++ (all versions)

The **MarshallSoft Client Mailer** can be used with the following Microsoft C/C++ compilers:

- **Microsoft Developer Studio** (VC 4.0)
- **Microsoft Visual Studio** (VC 5.0)
- **Microsoft Visual Studio** (VC 6.0)
- **Microsoft Visual Studio 2003, 2005, 2008 and 2010** (VC++ 7.0, VC++ 8.0, VC++ 9.0, VC++ 10.0)
- **Microsoft C++ Express Edition** (VC++ 9.0)

Visual C/C++ programs always link with MCM32.LIB.

2.2 Borland C/C++

The **MarshallSoft Client Mailer** can be used with the following Borland C/C++ compilers:

- **Borland C/C++** (Version 5.0)
- **Borland C/C++** (Version 5.5)
- **Borland C++ Builder** (Version 4.0 & up)

Borland programs always link with MCM32BCB.LIB. Borland is owned by Embarcadero Technologies.

2.3 Borland C++ Builder

- C++ Builder (version 4.0 to 2010)

Borland programs always link with MCM32BCB.LIB. Borland is owned by Embarcadero Technologies.

2.4 Watcom C/C++

The **MarshallSoft Client Mailer** can be used with the following Watcom C/C++ compilers:

- **Watcom C/C++** (Version 11.0)
- **Watcom C/C++** (Open Watcom)

Watcom programs always link with MCM32.LIB.

2.5 LCC-Win32 C

- LCC-Win32 (all versions)

LCC programs always link with MCM32.LIB.

2.6 MinGW GCC

- GCC (all versions)

GCC programs always link with MCM32.LIB.

3 TestMCM Example Program

The **TestMCM** example program verifies that Windows can find MCM32.DLL at runtime, and it displays the MCM version and related information.

4 The SendMail. Example Program

The **SendMail** example program sends a personalized letter to each recipient. The logic is:

```
Step 1: Read configuration parameters from SendMail.ini
Step 2: Attach MarshallSoft Client Mailer
Step 3: Open the letter to be sent
Step 4: Open the list of recipients
Step 5: Open skip lists
Step 6: Connect to SMTP server
      LOOP: Step 7: Merge letter with next recipient
            Step 8: Send the mail
Step 9: Wait for all channels to complete
Step 10: Close connection on all channels
```

If you are not familiar with the above logic, read the MCM Tutorial **mcm_tutorial.pdf** and the MCM Users Manual **mcm_users.pdf**.

The **SendMail** example program is very straight-forward and extensively documented. Open SendMail.c with any text editor and look through the code. Also open the SendMail.ini file and note the runtime parameters specified.

Edit SendMail.ini with your server name and (for authenticated logins) user name and password. Then compile and run **SendMail** from the Windows command prompt. To use the example letter (letter.txt), recipient list (list.txt), and specifying SendHistory.txt as the history file, using one channel and the parameters defined in SendMail.ini, type (from the command line)

```
SendMail 1 letter.txt list.txt SendHistory.txt
```

To merge mail, but not actually send it, uncomment the line (in SendMail.c)

```
#define SENDMAIL_MERGE_ONLY
```

To allow duplicate email addresses in the recipient list, uncomment the line

```
#define ALLOW_DUPLICATE_ADDRESSES
```

To allow empty fields in the recipient list, uncomment the line

```
#define ALLOW_EMPTY_FIELDS
```

If you will be connecting to a SMTP server that requires SSL, uncomment the line

```
#define ENABLE_SSL
```

If your security software scans outgoing email, you may need to disable this feature since some security software does not handle multiple connections correctly.

5 The GetReply. Example Program

The **GetReply** example program reads server and recipient replies. The logic is:

```
Step 1: Read configuration parameters from GetReply.ini
Step 2: Attach MarshallSoft Client Mailer
Step 3: Open bounced email file
Step 4: Open reply files
Step 5: Connect to the POP3 server
      LOOP: Step 6: Read next reply
Step 7: Close connection to server
```

Delete Codes

GetReply divides incoming email into three mutually exclusive classes:

1. Failure messages ("bounced email") from the server.
2. User reply message (e.g., "REMOVE email-address").
3. Everything else.

As each email is read when `mcmReadReply` is called

```
function mcmReadReply(SubjectBuffer, BufferSize, DeleteCode)
```

the 'DeleteCode' argument specifies if an email is to be deleted after it is read.

<u>DeleteCode</u>	<u>Action</u>
1	Delete if message is a failure message.
2	Delete if message is a recipient reply message.
4	Delete if message is not one of the above.

The above DeleteCodes can be added together to expand the messages deleted. For example, passing DeleteCode = 3 is equivalent to passing DeleteCode = 1 and DeleteCode = 2 separately. Hence, DeleteCode = 3 will delete failure (bounced) email and recipient replies but keep all other messages.

If no messages are to be deleted, use DeleteCode = 0

The **GetReply** example program is very straight-forward and extensively documented. Open `GetReply.c` with any text editor and look through the code. Also open the `GetReply.ini` file and note the runtime parameters specified.

Edit `GetReply.ini` with your server name. Then compile and run **GetReply**. To read server and client replies, using the parameters defined in `GetReply.ini`, type (from the command line)

```
GetReply
```

6 Compiling Example Programs.

There are three example programs: one to display the current version of MCM (**TestMCM**), one to send mail (**SendMail**) and one to read reply messages (**GetReply**). All are 32-bit Windows console mode programs designed to run from a command window.

6.1 Compiling TestMCM

The TestMCM example program can be compiled from either the command line or from the compiler's Integrated Development Environment (IDE).

6.1.1 Compiling From the Command Line

```
Visual C (4/5/6) : nmake TestMCM._m_  
Borland C (5.0) : make -f TestMCM._b_  
Borland C (5.5) : make -f TestMCM._i_  
Watcom (11.0)   : wmake -f TestMCM._w_  
MinGW          : TestMCM $GCC.bat
```

6.1.2 Compiling from IDE

```
Visual C (4/5/6) : TestMCM.mak  
Visual C (6)    : TestMCM.dsp  
Visual Studio   : TestMCM.vcproj  
Visual Studio 2008 : TestMCM(VS2008).vcproj  
Visual Studio 2010 : TestMCM(VS2010).vcxproj
```

6.2 Compiling SendMail

The SendMail example program can be compiled from either the command line or from the compiler's Integrated Development Environment (IDE).

6.2.1 Compiling From the Command Line

```
Visual C (4/5/6) : nmake SendMail._m_  
Borland C (5.0) : make -f SendMail._b_  
Borland C (5.5) : make -f SendMail._i_  
Watcom (11.0)   : wmake -f SendMail._w_  
MinGW          : SendMail$GCC.bat
```

6.2.2 Compiling from IDE

```
Visual C (4/5/6) : SendMail.mak  
Visual C (6)    : SendMail.dsp  
Visual Studio   : SendMail.vcproj  
Visual Studio 2008 : SendMail(VS2008).vcproj  
Visual Studio 2010 : SendMail(VS2010).vcxproj
```

6.3 Compiling GetReply

The GetReply example program can be compiled from either the command line or from the compiler's Integrated Development Environment (IDE).

6.3.1 Compiling From the Command Line

```
Visual C (4/5/6) : nmake GetReply._m_  
Borland C (5.0) : make -f GetReply._b_  
Borland C (5.5) : make -f GetReply._i_  
Watcom (11.0)   : wmake -f GetReply._w_  
MinGW          : GetReply$GCC.bat
```

6.3.2 Compiling from IDE

```
Visual C (4/5/6)   : GetReply.mak  
Visual C (6)      : GetReply.dsp  
Visual Studio     : GetReply.vcproj  
Visual Studio 2008 : GetReply(VS2008).vcproj  
Visual Studio 2010 : GetReply(VS2010).vcxproj
```

7 Revision History

Version 1.0: June 4, 2010.

- The official release of version 1.0.

Version 2.0: January 6, 2011

- Added MCM_GET_TIME_STAMP to mcmGetString().
- Call mcmOpenList(NULL) or mcmOpenList("") to rewind recipient list.
- mcmOpenLetter, mcmOpenList, mcmOpenBounce, mcmOpenReply, mcmOpenSkip allow both filenames and pathnames to be passed.
- Added mcmSetString(MCM_SET_CC_ADDRESS) & mcmSetString(MCM_SET_BCC_ADDRESS).
- Added MCM_GET_AUTH_VERS_MAJOR, MCM_GET_AUTH_VERS_MINOR, & MCM_GET_CID.
- Added MCM_SET_DUPLICATE_DETECT and MCM_SET_SERVER_TIMEOUT.
- The DLL runs indefinitely (does not stop working after one year).
- Maintenance updates are free.
- Activation Code Server allows up to 3 concurrent users with one activation file.
- Added support for LCC-WIN32 and MinGW GCC C compilers.

Version 3.0: May 19, 2011

- Removed requirement for activation code!
- Added mcmUtility function.
- Maximum channels increased to 32.
- Fixed problem: GetReply handles email addresses with trailing whitespace.
- Added MCM_GET_CUSTOMER_ID.
- Added MCM_GET_ALLOWED_CHANNELS.
- Added MCM_GET_ALLOWED_LIST_SIZE.
- Added MCM_GET_ALLOWED_SKIP_FILES.
- Added MCM_GET_ALLOWED_REPLY_FILES.
- Added MCM_GET_EDITION.
- Added MCM_GET_REGISTRATION.
- Allow tilde ~ and carot ^ as delimiters in recipient list file.

Version 3.1: January 11, 2012

- Added "UNSOLICITED BULK EMAIL" as bounce subject when receiving mail.
- Increased (macro delimiter) field size from 64 to 256 characters.
- Added MCM_ALLOW_EMPTY_FIELDS. Allows recipient list fields to be empty.
- Allow insertion of files by use of @@filename in recipient list file.
- Set letter substitution delimiter to backquote or backslash with mcmSetInteger(MCM_SET_MACRO_DELIMITER, delimiter).
- Example: mcmSetInteger(MCM_SET_MACRO_DELIMITER, `).
- Added mcmSetProxySSL(), which allows MCM to connect to servers requiring SSL.
- Added new merge code MCM_STAT_WITH_EMPTY_MACRO_STRING macro (1st list line) is empty.
- Added new merge code MCM_STAT_WITH_EMPTY_LIST_STRING entry in recipient list is empty.
- Added new stat code MCM_STAT_WITH_EMPTY_LIST_STRING counts "MCM_STAT_WITH_EMPTY_LIST_STRING" errors.