

MarshallSoft Client Mailer Library for C/C++

Programmer Manual

(MCM4C)

Version 6.0

May 10, 2021

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

Copyright (C) 2021
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	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 11
6.3	Compiling GetReply	Page 12
6.4	Compiling MakeStyled	Page 13
7	Revision History	Page 14

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** runs under 32-bit and 64-bit Windows through Windows 10. The **MarshallSoft Client Mailer DLLs (MCM32.DLL/MCM64.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. Both Win32 and Win64 DLLs are provided.

Since all programming language versions (C/C++, Visual Basic, Delphi, FoxPro, dBase, and XBase++) of MCM use the identical DLLs (MCM32.DLL or MCM64.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.

Details about the **TestMCM**, **SendMail** and **GetReply** example programs are provided in Sections 3, 4, and 5, respectively.

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.
- Includes **TestMCM** example program (with source) to verify installation and provide version information.
- Implemented as a standard Windows DLL, which will work with all versions of Windows.
- Supports both 32-bit and 64-bit Windows and includes Win32 and Win64 DLLs.
- Works with 32-bit and 64-bit Windows through Windows 10.
- License includes one year of technical support and downloadable updates for the MCM DLL.
- Royalty free distribution (no run-time fees) with your compiled application
License covers all programming languages (MCM4C, MCM4D, MCM4VB, MCM4FP, MCM4DB, MCM4XB).
- Works with all versions of Microsoft Visual C++ (v4.0 through Visual Studio 2015).
- Works with Borland C/C++ (v5.0, v5.5) and Borland C++ Builder.
- Works with Microsoft Foundation Class, Watcom v11, MinGW and LCC-WIN32.

1.2 Documentation Set

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

- **Tutorial Manual:** Introduces the basic functionality and overview of the **MarshallSoft Client Mailer**.
- **Servers Manual:** Covers background information on SMTP & POP3 servers.
- **User Manual:** Covers information that is not programming language specific (letter & list preparation, program logic, purchasing, performance, SSL, etc.).
- **Reference Manual:** Contains details for each individual MCM function specific for each programming language (C/C++, VB, etc.).
- **Programmer Manual:** Contains programming language (C/C++, VB, etc.) specific information such as compiling and running example programs.

The manuals can be found in the DOCS subdirectory/folder in the **MarshallSoft Client Mailer** file structure when it is installed. Note that for clarity purposes some information may have been provided in more than one manual.

- **Tutorial Manual:** mcm_tutorial.pdf or online at
http://www.marshallsoft.com/mcm_tutorial.pdf.
- **Servers Manual:** mcm_servers.pdf or online at
http://www.marshallsoft.com/mcm_servers.pdf.
- **User Manual:** mcm_users.pdf or online at
http://www.marshallsoft.com/mcm_users.pdf.
- **Reference Manuals** mcm4c_reference.pdf or
online at http://www.marshallsoft.com/mcm4c_reference.pdf.
- **Programmer Manuals** mcm4c_programmer.pdf or
online at http://www.marshallsoft.com/mcm4c_programmer.pdf.

All documentation can also be accessed online at <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"
- Letter header files begin with "Letter" and end with ".hdr"
- 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	32-bit MCL Dynamic Link Library (DLL) file.
mcm64.dll	64-bit MCL Dynamic Link Library (DLL) file
mcm32.lib	32-bit MCM Library (LIB) file.
mcm64.lib	64-bit 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 MCM4C52.ZIP using any Windows unzip program.

(3) Run the installation program SETUP.EXE which will install all MCM4C files, including copying MCM32.DLL and MCM64.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 and MCM64.DLL
SSL	- SSL proxy server files

1.6 Uninstalling

Uninstalling MCM4C is very easy. First, delete MCM32.DLL and MCM64.DLL that can be found in the \WINDOWS directory, typically C:\WINDOWS for Windows XP through Windows 10.

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

1.7 Purchase

Refer to the MCM User's Manual (**mcm_users.pdf** or online at http://www.marshallsoft.com/mcm_users.pdf) for detailed purchasing information.

MCM4C can be purchased 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.

For Win64, link with mcm64.lib rather than mcm32.lib

1.9 KeyCode

The MCM DLLs, MCM32.DLL and MCM64.DLL, each have a keycode encoded within them. The keycode is a 9 or 10-digit decimal number (unless it is 0), and will be found in the file KEYCODE.H. The keycode for the evaluation version is 0. The developer will receive a new key code after registering. The KEYCODE (NOT your Customer Id) is passed to **mcmAttach**.

If an error code (a negative number) is returned when calling **mcmAttach**, it means that the keycode in the MCM application does not match the keycode in the DLL (error -952), or that the evaluation version has expired (error -953).

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, 5.0, 6.0.**
- **Microsoft Visual Studio 2003, 2005, 2008, 2010, 2012, and 2013.**
- **Microsoft C++ Express Edition**

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/MCM64.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 User's 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 header file (letter.hdr), letter file (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.hdr 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
```

MCM can create a text file consisting of all recipients that were (attempted to be) sent and the corresponding server response. To create this file (at runtime), uncomment the line:

```
#define SET_RCPT_TRACE_FILES
```

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 Studio      : nmake TestMCM32._M_    (32-bit app)
                      nmake TestMCM64._M_    (64-bit app)

Visual C (4/5/6)   : nmake TestMCM32._M_
Borland C (5.0)    : make -f TestMCM32._B_
Borland C (5.5)    : make -f TestMCM32._I_
Watcom (11.0)      : wmake -f TestMCM32._W_
MinGW              : make -f TestMCM32._G_
                      TestMCM$GCC.bat
LCC-Win32          : make -f TestMCM32._L_
                      TestMCM$LCC.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
Visual Studio 2012  : TestMCM(VS2012).vcproj
Visual Studio 2013  : TestMCM(VS2013).vcxproj
Visual Studio 2015  : TestMCM(VS2013).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 Studio      : nmake SendMail32._M_      (32-bit app)
                     nmake SendMail64._M_      (64-bit app)

Visual C (4/5/6)   : nmake SendMail32._M_
Borland C (5.0)    : make -f SendMail32._B_
Borland C (5.5)    : make -f SendMail32._I_
Watcom (11.0)      : wmake -f SendMail32._W_
MinGW              : make -f SendMail32._G_
                     SendMail$GCC.bat
LCC-Win32          : make -f SendMail32._L_
                     SendMail$LCC.bat
```

6.2.2 Compiling from IDE

```
Visual C (4/5/6)   : SendMail.mak
Visual C (6)        : SendMail.dsp

Visual Studio       : vc_SendMail.vcproj
Visual Studio 2008  : vc_SendMail(VS2008).vcproj
Visual Studio 2010  : vc_SendMail(VS2010).vcxproj
Visual Studio 2012  : vc_SendMail(VS2012).vcxproj
Visual Studio 2013  : vc_SendMail(VS2013).vcxproj
Visual Studio 2015  : vc_SendMail(VS2013).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 Studio      : nmake GetReply32._M_      (32-bit app)
                     nmake GetReply64._M_      (64-bit app)

Visual C (4/5/6)  : nmake GetReply32._M_
Borland C (5.0)   : make -f GetReply32._B_
Borland C (5.5)   : make -f GetReply32._I_
Watcom (11.0)     : wmake -f GetReply32._W_
MinGW             : make -f GetReply32._G_
                     GetReply$GCC.bat
LCC-Win32         : make -f GetReply32._L_
                     GetReply$LCC.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
Visual Studio 2012 : GetReply(VS2012).vcproj
Visual Studio 2013 : GetReply(VS2013).vcxproj
Visual Studio 2015 : GetReply(VS2013).vcxproj
```

6.4 Compiling MakeStyled

The **MakeStyled** 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 Studio      : nmake MakeStyled32._M_      (32-bit app)
                      nmake MakeStyled64._M_      (64-bit app)

Visual C (4/5/6)   : nmake MakeStyled32._M_
Borland C (5.0)    : make -f MakeStyled32._B_
Borland C (5.5)    : make -f MakeStyled32._I_
Watcom (11.0)      : wmake -f MakeStyled32._W_
MinGW              : make -f MakeStyled32._G_
                      MakeStyled$GCC.bat
LCC-Win32          : make -f MakeStyled32._L_
                      MakeStyled$LCC.bat
```

6.1.2 Compiling from IDE

```
Visual C (4/5/6)   : MakeStyled.mak
Visual C (6)        : MakeStyled.dsp

Visual Studio       : MakeStyled.vcproj
Visual Studio 2008  : MakeStyled(VS2008).vcproj
Visual Studio 2010  : MakeStyled(VS2010).vcxproj
Visual Studio 2012  : MakeStyled(VS2012).vcproj
Visual Studio 2013  : MakeStyled(VS2013).vcxproj
Visual Studio 2015  : MakeStyled(VS2013).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.

Version 4.0: September 27, 2012

- Added macro delimiters tilde ~, caret ^, and backquote `.
- Ignore Tab chars on list macro line unless tab is the delimiter character.
- Added MCM_GET_LAST_EMAIL_SENT to mcmGetString().
- Moved letter headers to separate header file (e.g.: letter.hdr).
- Added mcmOpenHeader() function to open header file.
- Allow comments (line starting with #) in header section of letter header files.
- Changed default macro delimiter from % (percent) to ` (backquote)
- Allow recipient list to have a single entry (no delimiters)
- Added "Header: string" to headers. Used to specify a user header.
- Fixed problem in mcmRelease - was sometimes releasing SEE prematurely.
- Option to automatically load (letter) header file: mcmSetInteger(MCM_AUTO_LOAD_HEADER_FILE, 1)

Version 5.0: October 20, 2014

- Added 64-bit DLL (MCM64.DLL).
- Added project files for VS2012 and VS2013.
- Fixed problem in which not all reply strings (i.e., "REMOVE") were being detected.
- Fixed problem in which comments were counted as recipients in recipient list.
- Added MCM_SMTMP_ONLY and MCM_POP3_ONLY error messages
- Increased MRG_STRING_SIZE from 5000 to 65536 (64K) characters.
- Removed edition codes (no limit on list size).
- Added MCM_GET_NBR_EMAILS_DELETED to mcmGetInteger().
- Fixed problem in which incoming emails were not deleted as requested.
- Added MCM_ADD_BOUNCE_STRING to mcmSetString (adds 'bounce' subject)
- Added mcmGetString2()
- Added MCM_GET_BOUNCE_STRING to mcmGetString2()
- Fixed attachment problem when using two lists (1st w/attachment, 2nd w/o)
- Updated to SEE version 7.2.6

Version 5.1: September 16, 2016

- Fixed buffer overflow problem.
- Strings exceeding limits in recipient list are detected.
- Fixed problem with accumulating CC addresses.
- Updated to SEE version 7.4.1
- Additional details written to log file
- Added MCM_SET_RCPT_TRACE_FILE to mcmSetString() to generate file of server commands & responses.
- Added MCM_GET_LAST_RESPONSE to mcmGetString() to get last email server response.
- mcmRelease forces all files closed.
- Allow vertical bar '|' to specify multiple addresses in recipients file, as for example
 - mailAddress, CCAddress, Attachments
 - mike@mike4, pam@mike4|lauren@mike4, widget_red.txt
- Time of day added to "Today is" string written to log file
- Added new HTML templates.
- Added makefiles for MinGW GCC compiler
- Added makefiles for LCC-Win32 compiler
- Added function mcmSearch() that searches skip files for a specified string.

Version 5.2: March 15, 2019

- Warnings written if MCM_LOG_OFF was selected.
- Added diagnostics to merge code.
- Added MCM_NOT_TEXT_FILE error code (as @@ files are expected to be text files).
- Fixed problem in mcmRelease in which slot 0 not freed.
- Allow '#' comments after entries in recipient macro list file.
- Updated see32.obj and see64.obj to SEE version 8.0.4.
- Writes SEE version & build to MCM log file.
- Added mcmMakeStyled() function that creates an HTML file from an ASCII text file.
- Added MakeStyled example program that calls function mcmMakeStyled().
- Added MCM_CANNOT_CREATE_FILE error code.