

MESSAGE COMPILER 5.000

Mark Wilden
Sierra On-Line, Inc.

This manual describes the Message Compiler program, MC. The Message Compiler converts a text file into a message file suitable for use in the Message Editor and in Sierra SCI games.

Any text file can be compiled. To prepare a file for compilation, you simply add codes identifying the noun, verb, case, case comment, talker, message text, the comment text and the file number. It shouldn't be necessary to move text around; just add codes to what you've already got. Any part of the file that is not preceded by a Message Compiler code is ignored.

This manual is updated regularly and is stored in ASCII format with the Message Compiler in X:. See the last section "Changes" for a revision history.

Starting the Message Compiler

The syntax to run the Message Compiler is:

```
MC [-o] [-q] [-tXXX] [-u] [-h] [-?] filespec[.TXT] [filespec...]
```

where brackets indicate optional items. You may include wild cards in the **filespecs**. If you don't give an extension, **.TXT** is assumed. The Message Compiler compiles the files in the **filespecs** to similarly named **.MSG** files. If a given **.MSG** file already exists, you will be asked if you want to overwrite the file. If you want the output file to have a different extension, use the **-t** option to specify it. The **-o** option overwrites files without asking for confirmation. The **-q** option allows using quotes to delimit message text. The **-u** option replaces underscores in message text with spaces. The **-h** and **-?** flags display brief help.

Output file name

Unless you specify otherwise by means of the **##** code, the output file name is derived from the input file name by using the numbers in the input file name. For example, an input file **RM120.SC** will produce the output file **120.MSG**. If the name contains no numbers and there is no **##** code in the file, the output file will be named **0.MSG**. The extension is **.MSG** unless you specify otherwise with the **-t** command line option.

To explicitly set the output file name, include a code of the form

```
##xxx
```

before the first message in the file, where **xxx** represents a number from 0 through 46656. The output file will be called **XXX.MSG**. You can use this code to split an input file up into more than one output file. Whenever the **##** code is seen, the current file is written and a new file is started with the specified number. Output files are always overwritten, not appended to. Make sure you don't specify the same number twice in the same file, because the messages associated with the second use of the number will overwrite the messages associated with the first use.

Messages

A message is composed of: a noun, verb, case, sequence number, talker, message text, comment, and message number. Everything but the noun, talker and message text is optional. In addition, you can attach a case comment to the case. Nouns, verbs, cases and talkers are actually mnemonic identifiers that stand for numbers. The message number is an internal ID.

When the Message Compiler compiles an input file, it keeps track of the current noun, verb, case and talker so you don't have to identify them for each message. A message is added whenever a message text is seen, using the current noun, verb, case and talker, and using the comment text specified, if any. This means that there is no restriction on the order the noun, verb, case talker and the comment text appear in the file, but the message text must be last. Anything after the message text applies to the following message. The case comment must follow the case it applies to.

Other than the above ordering requirements, there are no other restrictions on the way your input files must be laid out. In particular, you can put message fields on different lines or on the same line.

Include files

Nouns, verbs, cases and talkers are stored in SCI include files to provide a link between mnemonic names you see in the Message Editor and the numbers that the SCI compiler expects. Nouns and cases are stored in **XXX.SHM**, where **XXX** represents the base file name of the associated message file. Verbs are stored in **VERBS.SH** and talkers are stored in **TALKERS.SH**. The **SINCLUDE** environment variable is used to find these files. New files will be created in the current directory if any of these files can't be found. Note that when these files are updated by the Message Compiler they are not locked as they are by the Message Editor.

Noun

To specify the noun for the following message(s), use the form

%%XXXXXXXXXXXXXXXXXX

where **XXXXXXXXXXXXXXXXXX** represents the noun name. The maximum length of a noun is 15 characters. When the Message Compiler sees the **%%** code, it reads characters into the noun name until a space, carriage return or ":" character is seen. If the name is too long, an error is reported. Names are converted to uppercase. These rules also apply to verbs, cases and talkers. When a noun is specified, the verb and case are reset; they no longer default to the last verb and case used. The maximum number of nouns per file is 255.

Verb

Verbs are specified with the form

>>XXXXXXXXXXXXXXXXXX

The maximum length is 15 characters. You don't have to specify the verb if it's the same as the last verb and the noun is the same as the last noun. When a verb is specified, the case is reset; it no longer defaults to the last case used. The maximum number of verbs per game is 255.

Case

Cases are specified with the form

==XXXXXXXXXXXXXXXXXX

The maximum length is 15 characters. You don't have to specify the case if it's the same as the last case and the noun and verb is the same as the last noun and verb. The maximum number of cases per file is 255.

Sequence number

Sequence numbers are specified with the form

++xx

where **xx** represents the sequence number. The maximum sequence number is 35. You will normally not need to use this field, since the Message Compiler generates sequence numbers automatically.

Talker

Talkers are specified with the form

`@@XXXXXXXXXX`

where `XXXXXXXXXX` represents the talker name. The maximum length of a talker name is 10 characters. Because of the preponderance of Narrator messages, the Narrator talker name may be abbreviated `@@N`. The maximum number of talkers per game is 255.

Message text

To specify the text for the message, use the form

```XXXXXX```

where `XXXXXX` represents the text of the message. The maximum length of message text is 2000 characters. Note that the ``` character is the back quote, located below the tilde ("`~`"), which is often near the top left of the keyboard. Also note that like all codes, you use two of these characters; don't confuse ```` with the `"` character.

If your input file already uses quotes ("`"`) to delimit messages, you may want to use the `-q` command line option when you run the Message Compiler. With this option, quotes can be used to delimit messages, as well as the usual ```` sequence. You must make sure that there are no extraneous quotes in your input file that might confuse the Message Compiler.

The following applies to message text, comment text and case comment text:

Certain special character sequences are allowed in text: `\n \r \xxx` (where `xxx` represents a decimal ASCII code). `\n` and `\r` both have the same effect: they emit a carriage return in the text. These characters are primarily for compiling SCI source code.

`\n` also has another purpose: to explicitly indicate carriage returns in input text. Normally, word wrapped input text has carriage returns at the end of every line. These carriage returns don't correspond to carriage returns the writer wants to see in the message output in the game (these are called "soft" carriage returns). Since there's no way for the Message Compiler to differentiate between soft carriage returns and the hard carriage returns the writer actually wants, it strips all carriage returns. Therefore, it's necessary to insert the `\n` character sequence where you want a hard carriage return to appear. Most messages don't have hard carriage returns, so this shouldn't prove too burdensome.

In order to correctly process text with left margins, all whitespace after a carriage return is deleted. If you use the `-u` command line option, an underscore character ("`_`") is converted to a space in message text (though not in comment or case comment text), in order to accommodate SCI source code conventions.

## Case comment

Case comments are specified with the form

**&&XXXXXXXXXXXXXXXX&&**

The maximum length is 1000 characters. The comment provides descriptive information for the case name. Each case comment applies to a single case; if you change the case comment for a case, it will be changed for all messages with that case in the given file.

## Comment text

To specify the comment text for the next message, use the form

**^^XXXXXX^^**

The maximum length of a comment is 2000 characters.

## Message number

Message numbers are generated by the Message Editor and should not be specified directly by you. They have the form

**||xxx**

where **xxx** represents the message number. When you print a message file to compiler format, these numbers are included. If you compile a source file that wasn't generated by the Message Editor, the Message Compiler will create new numbers.

## Reference noun

Reference nouns are like normal nouns, but have the form

**~noun**

## Reference verb

Reference verbs are like normal verbs, but have the form

**\$\$verb**

## Reference case

Reference cases are like normal cases, but have the form

**::case**

## Message Editor

The Message Editor has a provision for producing files that are compatible as input to the Message Editor. When you choose File|Print in the Message Editor, one of the options on the Print Destinations dialog is "Message Compiler output". Checking this box will print the file with Message Compiler codes inserted.

This lets you make large-scale changes to message files in your preferred word processor. For example, let's say you wanted to combine two message files into one. You'd do this by using the Message Editor to print the files with the "Message Compiler output" option checked. Then you'd combine the two files with your word processor. Then you'd compile the result into a new message file with the Message Compiler.

Whenever a message is created or changed in the Message Editor, the message is time-stamped so that messages that have changed between the original file and its translation can be determined. The Message Compiler gives all compiled messages the current date-stamp. This means that if there is a translated file, all messages will appear to have changed. For this reason, it is not advisable to use the Message Compiler to recreate message files if they have been translated. You can do this, but all messages will appear to be changed.

## CHANGES HISTORY

5.000	6/22/94	Use new resource type.
4.030	1/6/93	Made maximum sequence number 35.
4.020	11/20/92	Added -t option.
4.010	10/1/92	Fixed bug where the highest message number was reset to 0 after compiling a file. Allow duplicate or missing message numbers (as might happen after some cutting and pasting of the text file) by reassigning them new message numbers.
4.000	8/6/92	Supports ME 4 file format. Accepts   xxx format for message numbers. Skipped major version number to conform to file format version number.
2.030	6/18/92	When processing a list of files and an error occurs, allow user to continue processing remaining files. Added -u command line option.
2.022	6/16/92	Fixed bug where .SHM file wasn't saved if only the case comment changed.
2.021	6/4/92	Fixed bug where '\n' in an input message left a linefeed in the output message file and made the following character appear to be doubled.
2.020	3/26/92	Changed some limits: maximum noun, verb or case number is now 255; max module number is 46656; and max sequence number is 36.

- 
- |       |          |                                                                     |
|-------|----------|---------------------------------------------------------------------|
| 2.011 | 2/26/92  | Increased number of verbs to 255; decreased number of nouns to 255. |
| 2.010 | 2/19/92  | Improve default handling when verb, case or talker isn't specified. |
| 2.000 | 1/15/92  | Convert to new Message Editor format using noun-verb-case.          |
| 1.200 | 12/24/91 | Support -m option.                                                  |
| 1.100 | 12/6/91  | Support -q option. Added @@N abbreviation for Narrator.             |

