Input Message File Format

The input message file is a standard ASCII file containing three types of lines:

- Comment lines
- Component identifier line
- Component message lines

Comment Lines

Comments may appear anywhere in the file **except** between the component identifier line and the first message line. A comment line must begin with a semicolon (`;`) in the first column.

Example:

```
; This is a sample of an input
; message file for component DOS
; starting with three comment lines.
```

Component Identifier Line

A single line holding the three-character component identifier. This identifier is prefixed to all message numbers generated by MKMSGF.

In the example, the identifier is `DOS`.

Component Message Lines

Each message line consists of a **message header** followed by the ASCII text of the message.

Message Header

The header is composed of the following fields, written contiguously:

1. Three-character component identifier (e.g., `DOS`).
2. Four-digit message number (e.g., `0100`).
3. A single character indicating the message type (see table below).
4. A colon (':').
5. A single blank space separating the header from the message text.

Format: `COMPNNNNT: message text`

Message Types

Code	Type	Meaning
E	Error	Error message
H	Help	Help text
I	Information	Informational message
P	Prompt	Prompt (user input expected)
W	Warning	Warning message
?	(unused)	Placeholder for a number with no message text

Numbering and Placeholder Entries

- Message numbers may start at any value, but **must appear in strictly ascending sequential order** in the file.
- When a particular number is not used, a placeholder entry is required. A placeholder consists of the full header with the type ? and **no message text** after the colon and blank.

Example: `MAB0101?:`

Special Characters in the Message Text

The percent sign (%) has special meaning when used inside message text.

Suppressing the Trailing Newline (''%0'')

If the sequence `%0` appears as the very last characters of a message text, the `DosGetMessage` API will **not** append a carriage return and line feed (CR+LF). This is typically used for prompts so that user input can follow on the same line.

Inserting Variable Strings (''%1'' – ''%9'')

Sequences `%1` through `%9` mark positions where variable strings are inserted at runtime. The actual values are provided via the `Itable` and `IvCount` parameters of the `DosGetMessage` call.

Example: the message `%1 files copied` will have `%1` replaced by an appropriate string (e.g., a number of files).

Complete Input File Example

```
; This is a sample of an input
; message file for component MAB
; starting with three comment lines.
MAB
MAB0100E: File not found
```

```
MAB0101?:
MAB0102H: Usage: del [drive:][path] filename
MAB0103?:
MAB0104I: %1 files copied
MAB0105W: Warning! All data will be destroyed!
MAB0106?:
MAB0107?:
MAB0108P: Do you wish to apply these patches (Y or N)? %0
MAB0109E: Divide overflow
```

- `MAB` – component identifier.
- `MAB0100E` – error message.
- `MAB0101?` – placeholder for an unused number.
- `MAB0104I` – informational message with variable insertion (`%1`).
- `MAB0108P` – prompt with `%0` to suppress the trailing newline.

From:
https://osfree.su/doku/ - **osFree wiki**

Permanent link:
https://osfree.su/doku/doku.php?id=en:docs:tk:formats:msgsrc&rev=1779949251

Last update: **2026/05/28 06:20**