

message.txt Per Vim version 7.4. Ultima modifica: 2016 Sep 01

VIM REFERENCE MANUAL di Bram Moolenaar
Traduzione di questo testo: Antonio Colombo

Questo file contiene una lista di messaggi, di errore e non, prodotti da Vim. I messaggi sono listati in ordine alfabetico [...inglese -NdT]. Si può usare se il messaggio vi risulta poco chiaro. La lista è comunque incompleta.

1. Messaggi passati	:messages
2. Messaggi di errore	error-messages
3. Messaggi	messages

```
=====
1. Messaggi passati                                *:messages* *:mes* *message-history*
```

Il comando ":messages" si può usare per visualizzare i messaggi ricevuti in precedenza. Questo può tornare utile quando dei messaggi siano stati sovrascritti o troncati, a seconda del valore dall'opzione '**shortmess**'.

:messages	Elenca tutti i messaggi.
: {numero} messages	Elenca i {numero} messaggi più recenti.
:messages clear	Azzera l'elenco dei messaggi.
: {numero} messages clear	Azzera l'elenco dei messaggi, ma tiene gli ultimi {numero} più recenti.

Il numero di messaggi memorizzati è fissato in 20 per la versione minima e in 200 per tutte le altre versioni.

Il comando "g<" si può usare per rivedere l'ultima pagina dell'output di un comando eseguito in precedenza. Ciò è particolarmente utile quando si immette **<Spazio>** per sbaglio alla richiesta "Batti INVIO". Venite così riportati alla richiesta "Batti INVIO" da dove è possibile tornare ancora più indietro. Nota: Se avete interrotto l'output con "q" dopo un messaggio "-- Ancora --" verrà visualizzare solo l'output prodotto fino a quel punto. L'output dell'ultimo comando è perso quando un successivo comando produce dell'output. L'output del comando "g<" non è ridiretto.

Se state usando messaggi tradotti, la prima linea stampata vi dice chi mantiene i messaggi o la traduzione. L'indirizzo e-mail può permettervi di contattare il manutentore, se rilevate degli errori.

Per trovare aiuto riguardo a uno specifico messaggio (di errore o meno), usate l'identificativo posto all'inizio del messaggio. Per esempio per ottenere aiuto sul messaggio: >

E72: Close error on swap file

oppure (tradotto): >

E72: Errore durante chiusura swap file

Usate: >

:help E72

Se siete pigri, potete scrivere a lettere minuscole [ed abbreviare - NdT]: >

:h e72

```
=====
2. Messaggi di errore                                *error-messages* *errors*
```

Quando viene visualizzato un messaggio di errore, ma viene rimosso prima che

siate riusciti a leggerlo, lo potete rivedere ancora con: >

:echo errmsg

O vedere una lista di messaggi recenti con: >

:messages

Vedere **`:messages`** più sopra.

LISTA DI MESSAGGI

```
*E222* *E228* *E232* *E256* *E293* *E298* *E304* *E317*
*E318* *E356* *E438* *E439* *E440* *E316* *E320* *E322*
*E323* *E341* *E473* *E570* *E685* >
```

Aggiunto al buffer di lettura

makemap: modo non consentito

Non riesco a creare 'BalloonEval' con sia messaggio che callback

ERRORE processore Hangul

il blocco non era riservato

Non riesco a leggere blocco numero 0?

ml_upd_block0(): Non riesco a leggere blocco 0??"

ID blocco puntatori errato {N}

Aggiornati troppi blocchi?

ERRORE get_varp

u_undo: numeri linee errati

lista 'undo' non valida

linea di 'undo' mancante

ml_get: non riesco a trovare la linea {N}

Non riesco a trovare la linea {N}

numero linea non ammissibile: {N} dopo la fine

contatore linee errato nel blocco {N}

Errore interno

Errore interno: {funzione}

errore irreparabile in cs_manage_matches

Errore interno in Vim. Se riuscite a riprodurlo, siete pregati di inviare un rapporto sull'errore. Vedere |[bugs](#)|.

>

ATTENZIONE

Trovato uno swap file di nome ...

Vedere |[ATTENTION](#)|.

```
*E92* >
```

Buffer {N} non trovato

Il buffer richiesto non esiste. Questo messaggio può essere inviato anche dopo la cancellazione di un buffer contenente un mark o referenziato in qualche altro modo. |[:bwipeout](#)|

```
*E95* >
```

C'è già un buffer con questo nome

Non si possono avere due buffer con lo stesso nome.

```
*E72* >
```

Errore durante chiusura swap file

Lo |[swap-file](#)|, usato per tenere una copia del testo in modifica non è stato chiuso regolarmente. Di solito non c'è da preoccuparsi.

```
*E169* >
```

Comando troppo ricorsivo

Capita quando un comando Ex esegue un comando Ex che esegue un comando Ex, etc. Questo è permesso fino a 200 volte. Superata questa soglia, siamo di solito in presenza di un loop infinito. Probabilmente è in esecuzione un comando |[:execute](#)| o |[:source](#)|.

```
*E254* >
```

Non riesco ad allocare il colore {nome}

Il colore di nome {nome} è sconosciuto. SI veda |[gui-colors](#)| per una lista di colori disponibili sulla maggior parte delle piattaforme.

E458 >

Non riesco ad allocare elemento di colormap, possibili colori errati

Il messaggio indica che non ci sono abbastanza colori disponibili a Vim. Vim funziona lo stesso, ma alcuni colori non corrisponderanno alle specifiche. Provate a chiudere applicazioni che usano parecchi colori, o a farle partire dopo aver fatto partire gvim.

I browser sono noti consumatori di colori. Potete evitare l'inconveniente con Netscape chiedendogli di usare la sua mappa interna di colori: >

```
netscape -install
```

O ditegli di limitarsi a un dato numero di colori (64 dovrebbero bastare): >

```
netscape -ncols 64
```

La cosa si può anche definire nel vostro file Xdefaults: >

```
Netscape*installColormap: Yes
```

oppure >

```
Netscape*maxImageColors: 64
```

<

E79 >

Non posso espandere 'wildcard'

Un nome di file contiene una combinazione strana di caratteri che Vim tenta di espandere come **'wildcard'** (metacaratteri) senza riuscirci. Questo NON vuol dire che non ci siano nomi file corrispondenti all'espressione usata, ma solo che l'espressione stessa non è una espressione valida.

E459 >

Non posso tornare alla directory precedente

Durante l'espansione di un nome file Vim non riesce a tornare alla directory precedentemente usata. Tutti i nome file in uso potrebbero di conseguenza non essere più validi! È necessario (per usarla) di avere il permesso di esecuzione sulla directory corrente.

E190 *E212* >

**Non riesco ad aprire "{nome_file}" in scrittura
Non posso aprire il file in scrittura**

Per qualche ragione il file che state scrivendo non può essere creato o riscritto. Una ragione potrebbe essere che non siete autorizzati a scrivere nella directory o che il nome del file non è valido.

E166 >

Non posso aprire il file collegato ('linked') in scrittura

State tentando di scrivere un file che non può essere riscritto, e il file è un **'link'** (un 'hard link' [fra due file nello stesso File System - NdT] o un 'link simbolico' [fra due file nello stesso o in due diversi File System - NdT]). Una scrittura del file potrebbe ancora essere possibile, ma Vim non sa se volete eliminare il **'link'** all'altro file e scrivere il file come file separato, o se volete cancellare il file stesso e scrivere un nuovo file che lo sostituisca. Se veramente si vuole scrivere il file usando questo nome, occorre prima cancellare a mano il **'link'** o il file "collegato", oppure cambiare le autorizzazioni di accesso al file, in modo che Vim sia in grado di riscriverlo.

E46 >

Non posso cambiare la variabile read-only "{nome}"

Si sta tentando di assegnare un valore a un argomento di funzione |a:var| o a una variabile interna di Vim |v:var| in sola lettura ('read-only').

E90 >

Non riesco a scaricare l'ultimo buffer

Vim richiede sempre che ci sia un buffer da caricare, altrimenti non ci sarebbe nulla da visualizzare nella schermata.

E40 >

Non riesco ad aprire il file errori {nome_file}

Usando i comandi ":make" o ":grep": Il file usato per salvare i messaggi di errore o l'output del comando grep non può essere aperto. Ci possono essere

molti motivi per questo:

- **'shellredir'** è impostato in maniera incorretta.
- Lo shell cambia directory, e quindi il file di errore viene scritto in un'altra directory. Questo si può correggere cambiando **'makeef'**, ma così il comando viene ancora eseguito nella directory sbagliata.
- **'makeef'** è impostato in maniera incorretta.
- I programmi indicati in **'grepprg'** o **'makeprg'** non si sono potuti eseguire. Questo non è sempre facile da determinare (specialmente in MS-Windows). Controllare il vostro \$PATH.

>

Non riesco ad aprire il file C:\TEMP\{nome_file}.TMP

In MS-Windows, questo messaggio è visualizzato quando si intende leggere l'output di un comando, ma il comando non è stato eseguito con successo. Le cause possono essere molteplici. Controllare le opzioni **'shell'**, **'shellquote'**, **'shellxquote'**, **'shellslash'** e quelle ad esse collegate. Potrebbe anche darsi che il comando esterno non sia stato trovato, non c'è un messaggio di errore specifico per segnalare quest'ultima eventualità.

E12 >

Comando non ammesso da exrc/vimrc nella dir. in uso o nella ricerca tag

Alcuni comandi non sono permessi per motivi di sicurezza. Questi comandi provengono da un file .exrc o .vimrc nella directory corrente, oppure da un file di tag. Vedere anche **'secure'**.

E74 >

Comando troppo complesso

Una mappatura ha prodotto una stringa di comandi molto lunga. Potrebbe dipendere da una mappatura che indirettamente invoca se stessa.

>

ERRORE DI CONVERSIONE

Quando si sta scrivendo un file ed il testo "ERRORE DI CONVERSIONE" viene visualizzato, significa che alcuni bit sono andati persi nella conversione del testo dalla codifica UTF-8, utilizzata internamente da Vim, alla codifica usata nel file. Il file verrà marcato come "modificato" [cioè come se non fosse stato ancora salvato - NdT]. Se la perdita di informazione è rilevante, impostare l'opzione **'fileencoding'** ad un valore differente, in grado di gestire i caratteri nel buffer, e riscrivere ancora. Se invece la cosa non ha importanza si può abbandonare il buffer oppure resettare l'opzione **'modified'**.

E302 >

Non riesco a rinominare lo swap file

Quando il nome del file in modifica viene cambiato, Vim prova a rinominare anche il relativo **|swap-file|**. Se non ci riesce, continua ad usare il vecchio swap file. Messaggio quasi sempre innocuo.

E43 ***E44*** >

Stringa di confronto danneggiata
Programma 'regexp' corrotto

Qualcosa è andato storto all'interno di Vim, ed ha generato un'espressione regolare non valida. Se riuscite a riprodurre questo problema, per favore, inviateci un rapporto. Vedere **|bugs|**.

E208 ***E209*** ***E210*** >

Errore in scrittura di "{nome_file}"
Errore in chiusura di "{nome_file}"
Errore in lettura di "{nome_file}"

Vim sta tentando di rinominare un file, ma un semplice **'rename'** non basta. Allora si tenta di copiare il file, ma anche questo per qualche motivo non funziona. Il risultato può essere che sia il file d'origine che quello di destinazione esistono, ed il file di destinazione può risultare danneggiato.

>

Vim: Errore leggendo l'input, esco...

Vim non riesce a leggere dei caratteri immessi dall'utente, quando ha bisogno di input. Vim è bloccato, la sola cosa che può fare è terminare l'esecuzione. Può succedere quando sia **'stdin'** che **'stdout'** sono rediretti, durante l'esecuzione di uno **'script'**, il quale al suo interno non chiude Vim.

E47 >

Errore leggendo il file errori

Vim non è riuscito a leggere il file errori. La cosa NON è correlata con la mancata individuazione di un messaggio di errore.

E80 >

Errore in scrittura

La scrittura di un file non è andata a buon fine. Il file è probabilmente incompleto [dopo essere stato riscritto NdT].

E13 ***E189*** >

File esistente (aggiungi ! per riscriverlo)
"{nome_file}" esiste (aggiungi ! per eseguire comunque)

Vim protegge dalla riscrittura involontaria di un file, Se volete davvero scriverci sopra, usate lo stesso comando, ma aggiungete un **"!"** subito dopo il comando stesso.

Esempio: >

:w /tmp/test

diviene: >

:w! /tmp/test

<

E768 >

File swap esistente: %s (:silent! per sovrascriverlo)"

Protezione dal rischio di sovrascrivere un file in modifica con Vim. Questo capita quando usate **":w! nome_file"** ed esiste già uno swap file.

- Se lo swap file è un residuo di una sessione di Vim andata a male potreste voler cancellare lo swap file. Andate in edit su **{nome_file}** per ottenere informazioni a riguardo dello swap file.

- Se positivamente volete sovrascrivere il file, fatelo premettendo **":silent!"** al comando. Ad es: >

:silent! w! /tmp/test

< Il comando speciale è necessario, poiché era già stato aggiunto **!** per sovrascrivere un file esistente.

E139 >

File già caricato in un altro buffer

Tentativo di scrivere un file con un nome che è già in uso in un altro buffer. Così verreste ad avere due versioni dello stesso file [in modifica NdT].

E142 >

File non scritto: Scrittura inibita da opzione 'write'

L'opzione **'write'** è inibita. Per questo motivo tutti i comandi che tentano di scrivere un file generano questo messaggio. La cosa potrebbe dipendere da una opzione **| -m |** specificata alla partenza di Vim. Potete abilitare l'opzione **'write'** dando il comando **":set write"**.

E25 >

GUI non utilizzabile: Non abilitata in compilazione

La versione di Vim che state usando non contiene la parte GUI. Ragion per cui **"gvim"** e **":gui"** non sono disponibili.

E49 >

Quantità di 'scroll' non valida

Valore non permesso rilevato per le opzioni **'scroll'**, **'scrolljump'** o **'scrolloff'**.

E17 >

"{nome_file}" è una directory

Tentativo di scrivere un file con lo stesso nome di una directory esistente. Non è sensato. Probabilmente si deve aggiungere in fondo un nome file.

E19 >

'Mark' con numero linea non valido

Uso di un **'mark'** che indica un numero di linea inesistente. Ciò può accadere quando avete un **'mark'** che si riferisce ad un altro file, dal quale qualche altro programma ha tolto delle linee.

E219 *E220* >

Manca {.
Manca }.

Usando la notazione {} per indicare un nome file, c'è una "{" senza una corrispondente "}" o viceversa. Andrebbe usato così: **{foo,bar}**. Questo indica sia "foo" che "bar".

E315 >

ml_get: numero linea non valido: {numero}

Questo è un errore interno di Vim. Cercate di capire come riprodurlo, ed inviate un rapporto di errore. Vedere [|bugreport.vim|](#).

E173 >

ancora {numero} file da elaborare

Avete chiesto di uscire da Vim, ma almeno uno dei file sui quali avete chiesto di lavorare non è stato acceduto. Vim evita di uscire prima di aver finito di lavorare sui file richiesti. Vedere [|argument-list|](#). Se volete uscire lo stesso da Vim, ripetete il comando, che questa volta verrà eseguito.

E23 *E194* >

Nessun file alternato

Nessun nome file alternativo da sostituire a '#'

Il file alternativo non è ancora stato definito. Vedere [|alternate-file|](#).

E32 >

Manca nome file

Il buffer in uso non ha un nome. Per scriverlo, usate ":w nome_file". Oppure assegnate un nome al buffer con ":file nome_file".

E141 >

Manca nome file per il buffer {numero}

Uno dei buffer modificati non ha un nome file associato. Ragion per cui non può essere scritto. Dovete assegnare un nome file al buffer: >

```
:buffer {numero}
:file {nome_file}
```

<

E33 >

Nessuna espressione regolare precedente di 'substitute'

Se si usa il carattere '~' in una espressione regolare, questo viene sostituito dall'espressione precedentemente utilizzata in un comando ":substitute". Se non si è ancora usato il suddetto comando, la sostituzione non funziona. Vedere [|/~|](#). Questo capita anche usando ":s/pat/%/", dove il "%" indica la precedente stringa di sostituzione.

E35 >

Nessuna espressione regolare precedente

Specificando una espressione di ricerca vuota, viene usata l'ultima che era stata specificata. Se non si era fatta in precedenza alcuna ricerca, non ne esiste nessuna.

E24 >

Abbreviazione inesistente

Avete usato un comando ":unabbreviate" specificando un argomento che non è una

delle abbreviazioni definite. Tutte le variazioni di questo comando danno questo stesso messaggio: ":cunabbrev", ":iunabbrev", etc. Controllate se avete messo degli spazi alla fine della abbreviazione.

```
>
/dev/dsp: No such file or directory
```

Errore possibile solo usando la GUI GTK con supporto Gnome. Gnome tenta di usare un dispositivo audio, ma questo non è presente. Potete ignorare questo errore.

E31 >

Mapping inesistente

Avete usato un comando ":unmap" specificando un argomento che non è il nome di una mappatura definita. Tutte le variazioni di questo comando danno lo stesso messaggio: ":cunmap", ":unmap!", etc. Alcuni consigli:

- Controllate se avete messo degli spazi in fondo alla mappatura.
- Se la mappatura è locale ad un buffer, dovete usare ":unmap <buffer>".
|:map-<buffer>|

E37 *E89* >

Non salvato dopo modifica (aggiungi ! per eseguire comunque)

Buffer {numero} non salvato dopo modifica (aggiungi ! per eseguire comunque)

Avete chiesto di uscire senza salvare le modifiche |[abandon](#)| per un file che è stato modificato. Vim tenta di evitarvi di perdere il lavoro fatto. Potete scrivere il file modificato con ":w", o, se questa è la vostra intenzione, uscire senza salvare, |[abandon](#)|, perdendo tutte le modifiche. Ciò si ottiene aggiungendo un carattere '!' in fondo al comando che avete appena dato.

As esempio: >

```
:e un_altro_file
```

diventa: >

```
:e! un_altro_file
```

<

E162 >

Buffer "{nome}" non salvato dopo modifica

Il messaggio viene inviato se volete uscire da Vim, dopo aver modificato uno o più buffer. Dovete scrivere i buffer modificati (con |:w|), oppure usare un comando per ignorare le modifiche, ad es. ":qa!". Attenzione a non buttar via modifiche che invece intendevate tenere. Potreste esservi dimenticati di un buffer, specialmente se è stata impostata l'opzione '[hidden](#)' per nascondere.

>

[Non salvato dopo l'ultima modifica]

Questo messaggio appare quando si esegue un comando della shell mentre almeno uno dei buffer è marcato come modificato. Per evitare il messaggio, impostate a off l'opzione '[warn](#)'.

E38 >

Argomento nullo

Qualcosa è andato male all'interno di Vim, generando un puntatore invalido (NULL pointer). Se sapete come riprodurre l'errore, siete pregati di inviare un rapporto di errore. Vedere |[bugs](#)|.

E172 >

Ammesso un solo nome file

Il comando ":edit" accetta un solo nome di file. Se volete specificare più di un file da trattare, usate ":next" |:next|.

E41 *E82* *E83* *E342* >

Non c'è più memoria!

Non c'è più memoria! (stavo allocando {numero} byte)

Non riesco ad allocare alcun buffer, esco...

Non riesco ad allocare un buffer, uso l'altro...

Oh, oh. Stavate facendo qualcosa di complesso, oppure qualche altro programma sta consumando la memoria del computer. Attenzione! Vim non è in grado di gestire TUTTE le situazioni di mancanza di memoria. Per prima cosa

assicuratevi che le vostre modifiche siano state salvate. Solo dopo tentate di risolvere il problema di mancanza di memoria. Per andare sul sicuro, uscite da Vim, e ricominciate da capo.

I buffer sono tenuti in memoria sono in parte, e quindi la modifica di una file molto grosso non dovrebbe normalmente causare un problema di mancanza di memoria. L'informazione necessaria per l'undo è tenuta completamente in memoria, e potete ridurre la memoria usata con queste opzioni:

- **'undolevels'** Impostatelo a un valore basso, oppure a -1 per disabilitare completamente gli undo. Questo può servire se una modifica riguarda tutte le linee di un file.
- **'undoreload'** Impostatelo a zero per disabilitarlo.

E339 >

Espressione troppo lunga

Càpita con sistemi che usano 16 bit per rappresentare i numeri interi: il modello di espressione regolare compilato supera i 65000 caratteri circa di lunghezza. Provate a usare un'espressione regolare più breve. Càpita anche quando l'estensione di una regola non può essere contenuta nello spazio disponibile. Provate a semplificare l'espressione regolare.

E45 >

file in sola lettura (aggiungi ! per eseguire comunque)

State tentando di scrivere un file marcato come non modificabile, Per scrivere lo stesso il file, annullate l'opzione **'readonly'** oppure aggiungete il carattere '!' subito dopo il comando che avete dato. Ad es.: >

```
:w
va modificato in: >
:w!
```

<

E294 ***E295*** ***E301*** >

Errore leggendo swap file

Errore di posizionamento durante lettura swap file
Ahimè, lo swap file è perduto!!!

Vim ha provato a leggere del testo dallo `|swap-file|`, ma qualcosa non ha funzionato. Il testo nel buffer interessato può essersi corrotto! Controllate attentamente prima di riscrivere il buffer. Potreste preferire riscriverlo con un altro nome file, e controllare le differenze [rispetto al file di partenza].

E192 >

Uso ricorsivo di `:normal` troppo esteso

State usando un comando `":normal"`, il cui argomento usa un altro comando `":normal"` in maniera ricorsiva. La cosa è limitata a **'maxmapdepth'** livelli. Il seguente esempio mostra come produrre questo messaggio: >

```
:map gq :normal gq<CR>
```

Se digitate "gq", verrà invocata la mappatura, che invoca nuovamente "gq".

E22 >

Script troppo nidificati

File di comandi Vim si possono richiamare tramite l'argomento "-s" quando si manda in esecuzione Vim, e con il comando `":source"`. Questo file di comandi può a sua volta leggere un altro file di comandi. Questo può avvenire fino a un massimo di 14 livelli. Quando si supera tale soglia. Vim assume che si sia in presenza di un ciclo ricorsivo [errato - NdT] da qualche parte e si ferma con questo messaggio di errore.

E319 >

Spiacente, comando non disponibile in questa versione

Avete dato un comando non presente nella versione di Vim che state usando. Durante la compilazione di Vim, è possibile abilitare o escludere molte diverse componenti. Questo dipende da quanto grande si sceglie di generare Vim, come pure dal Sistema Operativo. Vedere `|+feature-list|` per stabilire se questa componente è disponibile. Il comando `|:version|` mostra con quali componenti Vim è stato compilato.

E300 >

Swap file already exists (symlink attack?)
Lo swap file esiste già (un link simbolico?)

Questo messaggio viene inviato quando vim prova ad aprire un file di swap che risulta esistere già, oppure che risulta essere un link simbolico. Questo non dovrebbe succedere, perché Vim ha già controllato che il file non esiste ancora. O qualcun altro ha aperto lo stesso file esattamente nello stesso momento (molto improbabile), oppure qualcuno sta tentando un attacco al vostro file tramite "symlink" (potrebbe capitare modificando un file nella directory /tmp oppure quando l'opzione '**directory**' inizia con "/tmp", che è una cattiva scelta).

E432 >

Tag file non ordinato alfabeticamente: {nome_file}

Vim (e Vi) si aspettano che i file di tag siano ordinati secondo la sequenza ASCII. Ciò permette di usare una ricerca binaria, molto più veloce che non quella lineare. Se i vostri file di tag non sono in questo ordine, reimpostate adeguatamente l'opzione |'**tagbsearch**'|. Questo messaggio viene visualizzato solo se Vim rileva dei problemi mentre sta cercando un tag. Talora il messaggio non viene inviato, anche se il file di tag non è correttamente ordinato.

E460 >

La 'fork' sulla risorsa verrebbe persa (aggiungi ! per eseguire comunque)

Nel Macintosh (classico), scrivendo un file, Vim tenta di mantenere ogni informazione relativa al file, compreso il suo "fork" sulla risorsa. Se ciò non è possibile, vedrete questo messaggio di errore. Aggiungete "!" al nome del comando per scrivere lo stesso (perdendo l'informazione).

E424 >

Troppi gruppi evidenziazione differenti in uso

Vim riesce a gestire solo circa 223 generi diversi di evidenziazione. Se superare questo limite, avete usato troppi comandi |:**highlight**| con diversi argomenti. I comandi ":highlight link" NON contribuiscono al totale.

E77 >

Troppi nomi file

Espandendo un nome file, si è trovata più di un nome che corrisponde. Un solo nome file è richiesto per il comando che era stato dato.

E303 >

Non riesco ad aprire lo swap file per "{nome_file}", recupero impossibile

Vim non è riuscito a creare un file di swap. Potete ancora modificare il file, ma se Vim termina in maniera imprevista, le modifiche andranno perse. Vim [lavorando solo in memoria NdT] consumerà un mucchio di memoria se modificate un grosso file. Potreste modificare l'opzione '**directory**' per evitare questo errore. Vedere |**swap-file**|.

E140 >

Usa ! per scrivere il buffer incompleto

Se si specifica un intervallo di linee per scrivere parte di un buffer, di solito non si intende ricoprire il file originale. Si tratta probabilmente di un errore (ad es. se la modalità Visuale era attiva e avete dato il comando ":w") e perciò Vim vi chiede di usare un ! dopo il comando, ad es.: ":3,10w!".

>

Warning: Cannot convert string "<Key>Escape,_Key_Cancel" to type VirtualBinding

Messaggi di questo tipo possono apparire quando si invoca Vim. Non si tratta di un problema di Vim, è la vostra configurazione X11 ad essere errata. Si possono trovare suggerimenti per risolvere di questo problema consultando: <http://groups.yahoo.com/group/solarisonintel/message/12179>.
 [questo URL non è più valido]

W10 >

Attenzione: Modifica a un file in sola-lettura

Il file è in sola lettura e voi lo state cambiando lo stesso. Potete usare l'autocomando `|FileChangedRO|` per evitare questo messaggio (l'autocomando deve cambiare l'opzione `'readonly'`). Vedere `'modifiable'` per impedire completamente qualsiasi cambiamento a un file. Questo messaggio è dato solo al primo cambiamento effettuato dopo che era stato impostato `'readonly'`.

W13 >

Attenzione: Il file "{nome_file}" risulta creato dopo l'apertura

State modificando con Vim un file che non esisteva prima, ma che è rilevato esistere ora. Dovete decidere se volete tenere la versione in modifica con Vim o il file creato nel frattempo. Questo messaggio non viene inviato quando `'buftype'` non è nullo.

W11 >

Attenzione: File "{nome_file}" modificato dopo l'apertura

Il file che avete cominciato a modificare è stato nel frattempo modificato ed il suo contenuto è cambiato (più precisamente: Se leggeste ancora il file con le impostazioni correnti di opzioni e di autocomandi, otterreste un testo differente). Questo probabilmente vuol dire che qualche altro programma ha modificato il file. A voi comprendere cosa è successo, e decidere quale versione del file volete conservare. Impostate l'opzione `'autoread'` se volete che questo avvenga automaticamente. Questo messaggio non viene inviato quando `'buftype'` non è nullo.

In un caso potete ricevere questo messaggio anche se non ci sono problemi: Se salvate il file in Windows nel momento in cui comincia l'ora legale. La cosa si può ovviare in uno dei modi seguenti:

- aggiungete questa linea al vostro autoexec.bat: >

SET TZ=-1

- < Modificate il "-1" a seconda del vostro fuso orario.
- Disabilitate "Passa automaticamente all'ora legale".
- Riscrivete ancora il file il giorno successivo. Oppure impostate la data del PC al giorno dopo, scrivete il file due volte, e tornate indietro con la data del PC.

W12 >

Attenzione: File "{nome}" modificato su disco ed anche nel buffer di Vim

Come il messaggio precedente, e il buffer per il file è modificato anche in Vim. Dovete decidere se tenere buona la versione in modifica con Vim o quella su disco. Questo messaggio non viene inviato quando `'buftype'` non è nullo.

W16 >

Attenzione: Modo File "{nome_file}" modificato dopo l'apertura

Se c'è stata una modifica che riguarda il modo (i permessi) del file, ma non il suo contenuto. Succede quando si richiede di modificare un file usando un software di controllo versione sorgenti, che reimposta il bit di "sola lettura". Non ci dovrebbero essere problemi a ricaricare il file. Impostate `'autoread'` se volete che il file sia ricaricato automaticamente.

E211 >

Il file "{nome_file}" non esiste più

Il file che avete cominciato a modificare è scomparso, o non è più accessibile. Scrivete il buffer altrove per evitare di perdere le modifiche. Questo messaggio non viene inviato quando `'buftype'` non è nullo.

W14 >

Attenzione: Superato limite della lista dei nomi di file

State usando veramente tanti buffer. È ora possibile che due buffer abbiano lo stesso numero, il che genera vari problemi. Sarebbe consigliabile uscire da Vim e ricominciare da capo.

E931 >

Non riesco a registrare il buffer

Memoria esaurita, oppure il numero di un buffer è duplicato. Può capitare dopo aver ricevuto un messaggio W14. Una ricerca nei buffer presenti potrebbe non funzionare sempre. Meglio far ripartire Vim.

E296 *E297* >

Errore di posizionamento scrivendo swap file
Errore scrivendo swap file

Questo succede per lo più quando non c'è spazio su disco. Vim non è riuscito a scrivere del testo nello `|swap-file|`. Questo non è pericoloso in sé, ma se Vim termina in maniera imprevista, una parte del testo può andare perduta, senza possibilità di recuperarla. Vim potrebbe anche esaurire la memoria disponibile al programma, laddove questo problema si ripresenti più volte.

connection-refused >

Xlib: connection to "<machine-name:0.0" refused by server

Questo succede quando Vim prova a connettersi al server X, ma il server X non permette la connessione. La connessione al server X è necessaria per poter modificare il titolo della sessione e per supportare la "clipboard" del terminale X (xterm). Sfortunatamente questo messaggio d'errore non è eliminabile, senza inibire le componenti `|+xterm_clipboard|` e `|+X11|`.

E10 >

\\ dovrebbe essere seguito da /, ? oppure &

Una linea comando inizia con un `"\"` o l'intervallo di linee di un comando ha un `"\"` in un punto sbagliato. Questo succede spesso se la continuazione di linee comando è disabilitata. Togliete la flag 'C' dall'opzione `'coptions'` per abilitarla. Oppure usate `":set nocp"`.

E471 >

Argomento necessario

Succede dopo aver eseguito un comando di Ex che richiede uno o più argomenti, ma non ne era stato specificato nessuno.

E474 *E475* >

Argomento non valido: {argomento}

È stato eseguito un comando di Ex, al quale era stato specificato un argomento non valido.

E488 >

Caratteri in più a fine comando

Un argomento è stato specificato per un comando Ex che non ne ammette.

E477 *E478* >

! non consentito
Non lasciarti prendere dal panico!

Avete aggiunto un `!"` ad un comando Ex che non ne ammette.

E481 >

Nessun intervallo consentito

Un intervallo è stato specificato per un comando Ex che non ne ammette. Vedere `|cmdline-ranges|`.

E482 *E483* >

Non riesco a creare il file {nome_file}
Non riesco ad ottenere nome file 'temp'

Vim non riesce a creare un file temporaneo.

E484 *E485* >

Non riesco ad aprire il file {nome_file}
Non riesco a leggere il file {nome_file}

Vim non riesce a leggere un file temporaneo. Specialmente in ambiente Windows, ciò può dipendere da una maniera errata di proteggere `[escape]` i caratteri speciali da inviare a `cmd.exe`; il metodo è stato cambiato dalla

patch 7.3.443. Provate a usare |shellescape()| per tutti gli argomenti passati allo shell tramite |system()|, oppure aggiungete esplicitamente la protezione con ^. Vedere anche 'shellxquote' e 'shellxescape'.

E464 >

Uso ambiguo di comando definito dall'utente

Esistono [almeno - NdT] due comandi definiti dall'utente con un prefisso comune, ed avete usato il completamento linea-Comando per eseguirne uno.

|user-cmd-ambiguous|

Esempio: >

```
:command MioComando1 echo "uno"
:command MioComando2 echo "due"
:MioComando
```

<

E492 >

Non è un comando dell'editor

Avete tentato di eseguire un comando che non è un comando Ex e neppure un comando definito dall'utente.

```
=====
3. Messaggi                                     *messages*
```

Questa è una panoramica (incompleta) di vari messaggi inviati da Vim:

```
*hit-enter* *press-enter* *hit-return*
*press-return* *hit-enter-prompt*
```

Batti INVIO o un comando per proseguire

Questo messaggio è inviato quando c'è qualcosa sullo schermo che dovrete leggere, prima che lo schermo venga ridisegnato:

- Dopo aver eseguito un comando esterno (ad es., ":!ls" e "=").
- Sulla linea di stato va visualizzato un messaggio più lungo della linea medesima, o che "invade" le aree contenenti messaggi 'showcmd' o 'ruler'.

- > Digitate <Invio> o <Spazio> per ridisegnare lo schermo e proseguire, senza che il tasto digitato venga usato in altro modo.
 - > Digitate ":" o qualunque comando in modalità Normale, per cominciare ad eseguire il comando stesso.
 - > Digitate 'k', <Up>, 'u', 'b' o 'g' per tornare indietro nei messaggi. Ciò funziona allo stesso modo che col |more-prompt| [-- Ancora --]. Disponibile solo quando 'compatible' vale off e 'more' vale on.
 - > La digitazione di 'j', 'f', 'd' o <Down> è ignorata quando i messaggi partono dalla cima dello schermo, 'compatible' vale off e 'more' vale on, per evitare che la digitazione di un 'j' o di un 'f' di troppo facciano scomparire tutti i messaggi.
 - > Digitate <C-Y> per copiare (yank - acciuffare) una selezione senza modalità nel registro di "clipboard".
 - > Usate un menù. I caratteri definiti per la modalità Linea-Comando sono usati.
 - > Quando l'opzione 'mouse' contiene la flag 'r', fare clic sul pulsante di sinistra del mouse equivale a digitare <Spazio>. Questo rende peraltro impossibile selezionare del testo.
 - > Per la GUI fare clic sul pulsante di sinistra del mouse sull'ultima linea è equivalente a premere <Spazio>.
- {Vi: solo i comandi ":" sono interpretati}

Se accidentalmente digitate <Invio> o <Spazio> e volete vedere il testo che era visibile in precedenza, usate |g<|. Questo funziona solo se 'more' è impostato.

Per ridurre il numero di richieste "Batti INVIO":

- Impostate 'cmdheight' a 2 o più.
- Aggiungete flag a 'shortmess'.
- Annullate opzioni 'showcmd' e/o 'ruler'.

Se un vostro script causa la richiesta "Batti INVIO" e non sapete perché, potete trovare utile la variabile |v:scrollstart|.

Vedere anche 'mouse'. Il messaggio "Batti INVIO" è evidenziato col gruppo sintattico |hl-Question|.

```

-- Ancora --
-- Ancora -- SPAZIO/d/j: schermo/pagina/riga giù, b/u/k: su, q: abbandona

```

Questo messaggio viene inviato se lo schermo è pieno di messaggi. È inviato solo quando l'opzione **'more'** è attivata. È evidenziato col gruppo `|hl-MoreMsg|`.

Digitate	effetto ~
<code><CR></code> o <code><NL></code> o j o <code><Giù></code>	una linea avanti
d	una pagina in giù (mezza videata)
<code><Spazio></code> o f o <code><PageDown></code>	una videata in giù
G	giù fino in fondo, ossia fino al messaggio "Batti INVIO"
<code><BS></code> o k o <code><Su></code>	una linea indietro (*)
u	mezza pagina in su (up) (*)
b o <code><PageUp></code>	pagina precedente (*)
q, <code><Esc></code> o CTRL-C	basta con la lista
:	basta con la lista ed immetti
	linea comando
<code><C-Y></code>	yank (copia) una selezione senza
	modalità alla "clipboard"
	(registri "*" e "+")
<code>{scelta-menu}</code>	quel che il menù definisce, in
	modalità Linea-Comando
<code><LeftMouse></code> (**)	pagina seguente

Ogni altro tasto provoca la visualizzazione del messaggio con le possibili scelte.

- (*) Tornare indietro è {non in Vi}. Il ritorno indietro è solo fin dove i messaggi cominciano a richiedere lo scorrimento.
- (**) Fare clic sul pulsante di sinistra del mouse funziona solo:
 - Per la GUI: nell'ultima linea dello schermo.
 - Se la flag 'r' è inclusa nell'opzione **'mouse'** (ma in questo caso la selezione di testo non funziona).

Nota: Il tasto digitato è ottenuto direttamente dal terminale, non è mappato ed eventuali altri caratteri immessi vengono ignorati.

Il comando `|g<|` si può usare per vedere l'ultima pagina dell'output di un comando eseguito in precedenza. Ciò è particolarmente utile quando si immette `<Spazio>` per sbaglio alla richiesta "Batti INVIO".

```
vim:tw=78:ts=8:ft=help:norl:
```

Per segnalazioni scrivere a vimdoc.it at gmail dot com
oppure ad Antonio Colombo azc100 at gmail dot com