



**EBook Gratuito**

# APPENDIMENTO

---

# debugging

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

**#debugging**

# Sommario

Di.....	1
<b>Capitolo 1: Iniziare con il debug</b> .....	<b>2</b>
Osservazioni.....	2
Examples.....	2
Esempio di debug della catena di strumenti per il debug in modalità utente su Windows.....	2
<b>Capitolo 2: Comprendere i codici di errore / uscita del sistema in Windows</b> .....	<b>3</b>
Osservazioni.....	3
Examples.....	3
Convertire i codici di uscita in messaggi significativi.....	3
<b>Titoli di coda</b> .....	<b>4</b>

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [debugging](#)

It is an unofficial and free debugging ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official debugging.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# Capitolo 1: Iniziare con il debug

## Osservazioni

Questa sezione fornisce una panoramica su cosa è il debug e perché uno sviluppatore potrebbe volerlo utilizzare.

Dovrebbe anche menzionare eventuali soggetti di grandi dimensioni all'interno del debug e collegarsi agli argomenti correlati. Poiché la Documentazione per il debug è nuova, potrebbe essere necessario creare versioni iniziali di tali argomenti correlati.

## Examples

### Esempio di debug della catena di strumenti per il debug in modalità utente su Windows

- Visual Studio (IDE)
- [WPA](#) (performance analyzer)
- [WinDbg](#) (debugger)
- [IDA Pro](#) (disassemblatore)
- [dotPeek](#) (decompilatore per .NET)
- [WinMerge](#) (strumento diff)
- [HxD](#) o [010 editor](#) (editor esadecimale)
- [Speedcrunch](#) (calcolatrice)
- [Firefox](#) (browser)
- [Monitor API Rohitab](#) (monitoraggio chiamate API)
- SOS (un'estensione WinDbg per .NET)
- [SOSex](#) (un'altra estensione per WinDbg)
- [Agent Ransack](#) (ricerca del contenuto del file)
- [WSCC](#) (raccolta di strumenti SysInternals e Nirsoft)
- [Debug Diag](#) (strumento di diagnosi)
- [Application Verifier](#) (strumento di analisi runtime)
- [Dipendente walker](#) (strumento di analisi delle dipendenze della DLL)
- [Stud\\_PE](#) (strumento di analisi dei file PE)

Leggi Iniziare con il debug online: <https://riptutorial.com/it/debugging/topic/5312/iniziare-con-il-debug>

---

# Capitolo 2: Comprendere i codici di errore / uscita del sistema in Windows

## Osservazioni

Perché questo processo funzioni, il codice di errore / uscita originale dovrebbe iniziare con `0x8007` che generalmente è un'indicazione che proviene da un processo Win32 valido.

Tuttavia, se non viene visualizzato alcun messaggio, probabilmente non ha avuto origine da un processo di Windows e pertanto, dovrà essere esaminato ulteriormente al di fuori dei passaggi sopra menzionati.

## Examples

### Convertire i codici di uscita in messaggi significativi

Essere in grado di capire i codici di errore / uscita è un'abilità fondamentale per gli sviluppatori sulla macchina di Windows. Eppure, per molti, il codice esadecimale criptico che può essere prodotto su un'applicazione che esce con errori può rivelarsi un processo lungo e faticoso da parte dello sviluppatore per rintracciare e isolare.

Ad esempio, su SO, ci sono diverse migliaia di domande tutte che chiedono il significato di cosa significhi un particolare codice di errore / uscita ... e come esempio, di seguito è riportato un tale codice di uscita

```
Il programma '[4432] program.exe' è terminato con il codice -2147023895  
(0x800703e9).
```

Pertanto, al fine di identificare la causa del problema, è necessario convertire il codice di uscita / errore in qualcosa di più significativo e possiamo farlo intraprendendo il seguente processo.

1. Dal codice di errore `0x800703e9` , prendi gli ultimi 4 caratteri `03e9`
2. Utilizzando un [convertitore da esadecimale a decimale](#) , converti `03e9` nella sua controparte decimale, che in questo caso è `1001`
3. Utilizzando `cmd` , digitare `net helpmsg 1001` o qualsiasi valore decimale viene restituito dal passaggio 3.
4. Dovrebbe apparire un messaggio di errore amichevole che può aiutare a identificare la causa del problema, che in questo caso, l'errore restituito era `Recursion too deep; the stack overflowed .`

Leggi [Comprendere i codici di errore / uscita del sistema in Windows online](#):

<https://riptutorial.com/it/debugging/topic/6146/comprendere-i-codici-di-errore---uscita-del-sistema-in-windows>

---

# Titoli di coda

S. No	Capitoli	Contributors
1	Iniziare con il debug	<a href="#">Community</a> , <a href="#">Daniel Nugent</a> , <a href="#">Thomas Weller</a>
2	Comprendere i codici di errore / uscita del sistema in Windows	<a href="#">timkly</a>