

# Raccolta personale



Developer Network





---

# Table Of Contents

## Capitolo 1

Informazioni introduttive sulla shell dei comandi

Cmd

Call

Date

For

Find

Fc

Findstr

Finger

Flattemp

Forfiles

Formato

Fsutil:

Ftp

Sottocomandi Ftp

Ftype

Shift

Utilizzo di parametri batch

Utilizzo dei filtri

---

# Capitolo 1

## Informazioni introduttive sulla shell dei comandi

Aggiornamento: gennaio 2005

Si applica a: Windows Server 2003, Windows Server 2003 R2, Windows Server 2003 with SP1, Windows Server 2003 with SP2

### Informazioni introduttive sulla shell dei comandi

La shell dei comandi è un programma software che consente di stabilire una comunicazione diretta tra l'utente e il sistema operativo. L'interfaccia utente non grafica della shell dei comandi rappresenta l'ambiente in cui eseguire applicazioni e utilità a caratteri. La shell dei comandi esegue programmi e ne visualizza l'output sullo schermo utilizzando singoli caratteri simili all'interprete dei comandi di MS-DOS, Command.com. La shell dei comandi del sistema operativo server Windows utilizza l'interprete dei comandi Cmd.exe, che carica le applicazioni e indirizza il flusso di informazioni tra di esse, per convertire l'input dell'utente in un formato comprensibile dal sistema operativo.

È possibile utilizzare la shell dei comandi per creare e modificare file batch, denominati anche script, con cui rendere automatiche le attività più comuni. Ad esempio, è possibile utilizzare script per automatizzare la gestione degli account utente o l'esecuzione dei backup notturni. È inoltre possibile utilizzare la versione per riga di comando del programma Windows Script Host, CScript, per eseguire script più complessi nella shell dei comandi. Per ulteriori informazioni, vedere [Eseguire Windows Script Host](#). L'utilizzo di file batch consente di eseguire operazioni in modo più efficiente rispetto a quanto possibile con l'interfaccia utente. I file batch infatti accettano tutti i comandi disponibili per la riga di comando. Per ulteriori informazioni sui file batch e sugli script, vedere [Utilizzo di file batch](#).

È possibile personalizzare la finestra del prompt dei comandi per migliorare la leggibilità dei dati e per esercitare un maggiore controllo sul modo in cui i programmi vengono eseguiti. Per ulteriori informazioni sulla personalizzazione della finestra del prompt dei comandi, vedere [Configurare il prompt dei](#)

comandi.

## Utilizzo della sintassi dei comandi

La sintassi indica in quale ordine digitare un comando e i relativi parametri. L'esempio che segue del comando **xcopy** illustra una serie di formati di testo della sintassi:

**xcopy** *Origine* [*Destinazione*] [/w] [/p] [/c] [/v] [/q] [/f] [/l] [/g] [/d[:MM-GG-AAAA]] [/u] [/i] [/s] [/e] [/t] [/k] [/r] [/h] [{/a | /m}] [/n] [/o] [/x] [/exclude:File1[+[File2]][+[File3]]] [{/y | /-y}] [/z]

Nella tabella seguente viene spiegato come interpretare i diversi formati di testo.

## Convenzioni di formattazione

Formato	Significato
Corsivo	Informazioni che devono essere specificate dall'utente
Grassetto	Elementi che l'utente deve digitare esattamente come indicato
Puntini (...)	Parametro che è possibile ripetere più volte in una riga di comando
Parentesi quadre ([])	Elementi facoltativi
Parentesi graffe ({}); possibili scelte separate da una barra verticale ( ). Esempio: {even odd}	Opzioni alternative tra le quali l'utente deve sceglierne una sola
Tipo di carattere Courier	Codice oppure output di programmi

## Utilizzo di più comandi e di simboli di elaborazione condizionale

È possibile eseguire più comandi da una singola riga di comando o script utilizzando simboli di elaborazione condizionale. Quando si eseguono più comandi con simboli di elaborazione condizionale, i comandi a destra del simbolo agiscono in base ai risultati del comando a sinistra del simbolo. Ad esempio, è possibile fare in modo che un comando venga eseguito solo se il comando precedente non riesce oppure solo se il comando precedente è stato eseguito correttamente.

I caratteri speciali riportati nella tabella che segue possono essere utilizzati per passare più comandi.

Carattere	Sintassi	Definizione
<b>&amp;</b> [...]	<i>Comando1 &amp;Comando2</i>	Da utilizzare per separare più comandi digitati sulla stessa riga di comando. Cmd.exe esegue il primo comando, quindi il secondo.
<b>&amp;&amp;</b> [...]	<i>Comando1 &amp;&amp;Comando2</i>	Da utilizzare per eseguire il comando che segue <b>&amp;&amp;</b> solo se il comando che precede il simbolo viene eseguito correttamente. Cmd.exe esegue il primo comando, quindi esegue il secondo solo se il primo è stato completato correttamente.
<b>  </b> [...]	<i>Comando1    Comando2</i>	Da utilizzare per eseguire il comando che segue <b>  </b> solo se il comando che precede questo simbolo non viene eseguito correttamente. Cmd.exe esegue il primo comando, quindi esegue il secondo solo se il primo non è stato completato correttamente (ha ricevuto un codice di errore superiore a zero).
<b>( )</b> [...]	<i>( Comando1 &amp; Comando2 )</i>	Da utilizzare per raggruppare o nidificare più comandi.
<b>;</b> <b>o</b> <b>,</b>	<i>Comando1 Parametro1;Parametro2</i>	Da utilizzare per separare i parametri dei comandi.

## Note

- La *e commerciale* (&), la *barra verticale o pipe* (|) e le *parentesi* ( ) sono caratteri speciali che devono essere preceduti dal carattere di escape (^) o da virgolette doppie quando vengono passati come argomenti.
- Se un comando completa un'operazione correttamente, restituisce un codice pari a zero o non restituisce alcun codice. Per ulteriori informazioni sui codici di uscita, vedere [Utilizzo dei Deployment Kit e dei Resource Kit di Windows](#).

## Nidificazione delle shell dei comandi

È possibile nidificare le shell dei comandi all'interno di Cmd.exe aprendo una nuova istanza di Cmd.exe dal prompt dei comandi. Per impostazione predefinita, ogni istanza di Cmd.exe eredita l'ambiente dell'applicazione Cmd.exe padre. La nidificazione di istanze di Cmd.exe consente di apportare modifiche all'ambiente locale senza conseguenze sull'applicazione padre di Cmd.exe. In questo modo l'ambiente originario di Cmd.exe viene mantenuto intatto ed è possibile tornarvi

dopo aver terminato la shell dei comandi nidificata. Le modifiche apportate nella shell dei comandi nidificata non vengono salvate.

Per nidificare una shell dei comandi, al prompt dei comandi digitare:

### **cmd**

Verrà visualizzato un messaggio simile al seguente:

```
Microsoft (R) Windows Server 2003, Standard Edition (TM)(C) Copyright 1985-2002 Microsoft Corp.
```

Per chiudere la shell dei comandi nidificata, digitare **exit**.

È possibile localizzare ulteriormente le modifiche in un'istanza di Cmd.exe o in uno script mediante i comandi **setlocal** ed **endlocal**. **Setlocal** crea un ambito locale mentre **endlocal** lo termina. Tutte le modifiche apportate all'interno dell'ambito **setlocal** ed **endlocal** vengono eliminate, lasciando quindi immutato l'ambiente originario. È possibile nidificare questi due comandi fino a un massimo di 32 livelli. Per ulteriori informazioni sui comandi **setlocal** e **endlocal**, vedere Setlocal e [Endlocal](#).

## **Utilizzo di variabili di ambiente con Cmd.exe**

L'ambiente della shell dei comandi di Cmd.exe è definito da variabili che determinano il funzionamento della shell dei comandi e del sistema operativo. È possibile definire il funzionamento dell'ambiente della shell dei comandi o dell'ambiente di tutto il sistema operativo utilizzando due tipi di variabili di ambiente, ovvero di sistema e locali. Le variabili di ambiente di sistema definiscono il funzionamento dell'ambiente del sistema operativo globale. Le variabili di ambiente locali definiscono il funzionamento dell'ambiente dell'istanza attuale di Cmd.exe.

Le variabili di ambiente di sistema sono preimpostate nel sistema operativo e sono disponibili per tutti i processi del sistema operativo server Windows. Solo gli utenti con credenziali di amministratore possono modificare le variabili di sistema. Tali variabili vengono generalmente utilizzate negli script per i processi di accesso.

Le variabili di ambiente locali sono disponibili solo quando l'utente per il quale sono state create è connesso al computer. Le variabili locali impostate nell'hive **HKEY\_CURRENT\_USER** sono valide solo per l'utente corrente, ma definiscono il funzionamento dell'ambiente del sistema operativo globale.

Nell'elenco che segue verranno descritti i vari tipi di variabili in ordine decrescente di precedenza:

1. Variabili di sistema incorporate
2. Variabili di sistema presenti nell'hive **HKEY\_LOCAL\_MACHINE**
3. Variabili locali presenti nell'hive **HKEY\_CURRENT\_USER**
4. Tutte le variabili di ambiente e i percorsi impostati nel file Autoexec.bat



5. Tutte le variabili di ambiente e i percorsi impostati in uno script di accesso, se presente

6. Variabili utilizzate in modo interattivo in uno script o in un file batch

Nella shell dei comandi ogni istanza di Cmd.exe eredita l'ambiente della rispettiva applicazione padre. È pertanto possibile modificare le variabili nel nuovo ambiente Cmd.exe senza conseguenze sull'ambiente dell'applicazione padre.

Nella tabella riportata di seguito sono elencate le variabili di ambiente di sistema e locali per il sistema operativo server Windows.

<b>Variabile</b>	<b>Tipo</b>	<b>Descrizione</b>
%ALLUSERSPROFILE%	Local	Restituisce il percorso del profilo Tutti gli utenti.
%APPDATA%	Local	Restituisce il percorso in cui le applicazioni memorizzano i dati per impostazione predefinita.
%CD%	Local	Restituisce la stringa della directory attuale
%CMDCMDLINE%	Local	Restituisce la riga di comando utilizzata per avviare il programma Cmd.exe attuale.
%CMDEXTVERSION%	Sistema	Restituisce il numero di versione delle estensioni del processore dei comandi attuale.
%COMPUTERNAME%	Sistema	Restituisce il nome del computer.
%COMSPEC%	Sistema	Restituisce il percorso esatto dell'eseguibile della shell dei comandi.
%DATE%	Sistema	Restituisce la data attuale. Utilizza lo stesso formato del comando <b>date /t</b> . Generato da Cmd.exe. Per ulteriori informazioni sul comando <b>date</b> , vedere <a href="#">Date</a> .
%ERRORLEVEL%	Sistema	Restituisce il codice di errore del comando utilizzato più di recente. Un valore diverso da zero generalmente indica un errore.

%HOMEDRIVE%	Sistema	Restituisce la lettera dell'unità locale della workstation dell'utente connessa alla home directory dell'utente. Impostato in base al valore della home directory. La home directory dell'utente è specificata in Utenti e gruppi locali.
%HOMEPATH%	Sistema	Restituisce il percorso completo della home directory dell'utente Impostato in base al valore della home directory. La home directory dell'utente è specificata in Utenti e gruppi locali.
%HOMESHARE%	Sistema	Restituisce il percorso di rete della home directory condivisa dell'utente. Impostato in base al valore della home directory. La home directory dell'utente è specificata in Utenti e gruppi locali.
%LOGONSERVER%	Local	Restituisce il nome del controller di dominio che ha convalidato la sessione di accesso attuale.
%NUMBER_OF_PROCESSORS%	Sistema	Specifica il numero di processori installati nel computer.
%OS%	Sistema	Restituisce il nome del sistema operativo. In Windows 2000 il sistema operativo viene visualizzato come Windows NT.
%PATH%	Sistema	Specifica il percorso di ricerca dei file eseguibili.
%PATHEXT%	Sistema	Restituisce un elenco delle estensioni dei file che il sistema operativo considera eseguibili.
%PROCESSOR_ARCHITECTURE%	Sistema	Restituisce l'architettura del chip del processore. Valori: x86 o IA64 (basata su Itanium).
%PROCESSOR_IDENTIFIER%	Sistema	Restituisce una descrizione del processore.
%PROCESSOR_LEVEL%	Sistema	Restituisce il numero di modello del processore installato nel computer.
%PROCESSOR_REVISION%	Sistema	Restituisce il numero di revisione del processore.
%PROMPT%	Local	Restituisce le impostazioni del prompt dei comandi per l'interprete attuale. Generato da Cmd.exe.
%RANDOM%	Sistema	Restituisce un numero decimale casuale compreso tra 0 e 32767. Generato da Cmd.exe.
%SYSTEMDRIVE%	Sistema	Restituisce l'unità contenente la directory principale del sistema operativo server Windows, ovvero la directory principale del sistema (system root).

%SYSTEMROOT%	Sistema	Restituisce il percorso della directory principale del sistema operativo server Windows.
%TEMP% e %TMP%	Sistema e utente	Restituisce le directory temporanee predefinite utilizzate dalle applicazioni disponibili agli utenti attualmente connessi. Alcune applicazioni richiedono TEMP, mentre altre richiedono TMP.
%TIME%	Sistema	Restituisce l'ora attuale. Utilizza lo stesso formato del comando <b>time /t</b> . Generato da Cmd.exe. Per ulteriori informazioni sul comando <b>time</b> , vedere <a href="#">Ora</a> .
%USERDOMAIN%	Local	Restituisce il nome del dominio contenente l'account dell'utente.
%USERNAME%	Local	Restituisce il nome dell'utente attualmente connesso.
%USERPROFILE%	Local	Restituisce il percorso del profilo dell'utente attuale.
%WINDIR%	Sistema	Restituisce il percorso della directory del sistema operativo.

## Impostazione delle variabili di ambiente

Utilizzare il comando **set** per creare, modificare, eliminare o visualizzare variabili di ambiente. Il comando **set** modifica le variabili solo nell'ambiente della shell corrente.

Per visualizzare una variabile, al prompt dei comandi digitare:

**set** *NomeVariabile*

Per aggiungere una variabile, al prompt dei comandi digitare:

**set***NomeVariabile=Valore*

Per eliminare una variabile, al prompt dei comandi digitare:

**set** *NomeVariabile* =

È possibile utilizzare molti caratteri come valori delle variabili, incluso lo spazio. I caratteri speciali <, >, |, & o ^, se utilizzati, devono essere preceduti da un carattere di escape (^) o da virgolette doppie. Se si utilizzano le virgolette, esse verranno incluse come parte del valore in quanto tutto quello che segue il segno di uguale viene considerato un valore. Si prendano in considerazione i seguenti esempi:

- Per creare il valore della variabile **nuovo&nome**, digitare:

**set varname=nuovo^&nome**

- Per creare il valore della variabile **"nuovo&nome"**, digitare:

**set varname="nuovo&nome"**

- Se si digita **set varname=nuovo&nome** al prompt dei comandi, verrà visualizzato un messaggio di errore simile al seguente:

```
"'nome' non è riconosciuto come comando interno o esterno, un programma eseguibile o un file batch."
```

Per i nomi delle variabili non è importante la distinzione tra maiuscole e minuscole. **set** tuttavia visualizza la variabile esattamente come è stata digitata. È possibile combinare lettere maiuscole e minuscole nei nomi delle variabili per rendere il codice più leggibile, ad esempio NomeUtente.

## Note

- La dimensione massima della singola variabile di ambiente è 8192 byte.
- La dimensione massima totale di tutte le variabili di ambiente, inclusi i nomi delle variabili e i segni di uguale, è pari a 65.536 KB.

## Sostituzione dei valori delle variabili di ambiente

Per consentire la sostituzione dei valori delle variabili dalla riga di comando o negli script, racchiudere il nome della variabile tra i simboli di percentuale, ovvero **%NomeVariabile%**. Utilizzando i simboli di percentuale, Cmd.exe fa riferimento ai valori delle variabili anziché effettuare un confronto letterale. Dopo aver definito i valori per un nome di variabile, racchiudere il nome della variabile tra i simboli di percentuale. Cmd.exe cercherà tutte le istanze del nome della variabile e sostituirà il nome con il valore definito per la variabile. Ad esempio, se si crea uno script contenente diversi valori, quali nomi utente, e si desidera definire la variabile di ambiente USERNAME per ciascun utente con questi valori, è possibile scrivere uno script utilizzando la variabile USERNAME racchiusa tra simboli di percentuale. Quando si esegue questo script, Cmd.exe sostituisce %USERNAME% con i valori della variabile, eliminando quindi la necessità di eseguire questa attività manualmente per ogni utente. La sostituzione delle variabili non è ricorsiva. Cmd.exe controlla le variabili una volta. Per ulteriori informazioni sulla sostituzione delle variabili, vedere [For](#) e [Call](#).

# Cmd

Aggiornamento: gennaio 2005

Si applica a: Windows Server 2003, Windows Server 2003 R2, Windows Server 2003 with SP1, Windows Server 2003 with SP2

## Cmd

Avvia una nuova istanza dell'interprete dei comandi Cmd.exe. Se utilizzato senza parametri, il comando **cmd** visualizza le informazioni sulla versione e sul copyright del sistema operativo.

Sintassi

**cmd** [{/c | /k}] [/s] [/q] [/d] [{/a | /u}] [/t:PS] [/e:{on | off}] [/f:{on | off}] [/v:{on | off}] [*Stringa*]

Parametri

**/c**

Esegue il comando specificato da *Stringa*, quindi interrompe l'esecuzione.

**/k**

Esegue il comando specificato da *Stringa* e continua l'esecuzione.

**/s**

Modifica il trattamento di *Stringa* dopo **/c** o **/k**.

**/q**

Disattiva l'eco.

**/d**

Disattiva l'esecuzione dei comandi di esecuzione automatica.

**/a**

Formatta l'output del comando interno in un pipe o in un file in base alle specifiche ANSI (American National Standards Institute).

**/u**

Formatta l'output del comando interno in un pipe o in un file in base alle specifiche Unicode.

**/t: PS**

Imposta i colori di primo piano *P* e di sfondo *S*. Nella tabella seguente sono riportati i valori esadecimali validi che è possibile utilizzare per *P* e *S*.

Valore	Colore
0	Nero
1	Blu scuro
2	Verde
3	Verde acqua
4	Bordeaux
5	Viola
6	Verde oliva
7	Grigio chiaro
8	Grigio
9	Blu
A	Verde limone

B	Azzurro
C	Rosso
D	Fucsia
E	Giallo
F	Bianco

**/e:on**

Abilita le estensioni ai comandi.

**/e:off**

Disabilita le estensioni ai comandi.

**/f:on**

Abilita il completamento dei nomi di file e directory.

**/f:off**

Disabilita il completamento dei nomi di file e directory.

**/v:on**

Abilita l'espansione differita delle variabili di ambiente.

**/v:off**

Disabilita l'espansione differita delle variabili di ambiente.

***Stringa***

Specifica il comando che si desidera eseguire.

**/?**

Visualizza informazioni della Guida al prompt dei comandi.

## Osservazioni

- Utilizzo di più comandi

Per *Stringa* è possibile specificare più comandi separati da **&&** racchiudendoli tra virgolette, ad esempio "*Comando&&Comando&&Comando*".

- Elaborazione delle virgolette

Se si specifica **/c** o **/k**, il comando **cmd** elabora la parte restante di *Stringa* e le virgolette vengono mantenute solo se sono soddisfatte le condizioni seguenti:

- Non si utilizza **/s**.
- Si utilizza un unico set di virgolette.
- Le virgolette non racchiudono caratteri speciali (ad esempio &<>()@^|).
- Si utilizza uno più spazi all'interno del testo tra virgolette.
- La *Stringa* racchiusa tra virgolette corrisponde al nome di un file eseguibile.

Se queste condizioni non sono soddisfatte, *Stringa* viene elaborata esaminando il primo carattere per verificare se si tratta di un segno di virgolette di apertura. In caso affermativo, il carattere viene rimosso unitamente al segno di virgolette di chiusura. L'eventuale testo che segue le virgolette di chiusura viene conservato.

- Esecuzione delle sottochiavi del Registro di sistema

Se non si specifica **/d** in *Stringa*, Cmd.exe eseguirà la ricerca delle sottochiavi del Registro di sistema seguenti:

**HKEY\_LOCAL\_MACHINE\Software\Microsoft\Command Processor\AutoRun\REG\_SZ**

**HKEY\_CURRENT\_USER\Software\Microsoft\Command Processor\AutoRun\REG\_EXPAND\_SZ**

Se una o entrambe le sottochiavi del Registro di sistema vengono trovate, queste vengono eseguite prima delle altre variabili.

### Attenzione



- Modifiche non corrette al Registro di sistema possono danneggiare gravemente il sistema. Si consiglia di effettuare il backup di tutti i dati più importanti presenti nel computer prima di procedere.

- Abilitazione e disabilitazione delle estensioni ai comandi

In Windows XP le estensioni ai comandi sono abilitate per impostazione predefinita. È possibile disabilitarle per un'operazione particolare utilizzando **/e:off**. È possibile abilitare o disabilitare le estensioni a tutte le opzioni della riga di comando di **cmd** in un computer o in una sessione utente impostando i valori di **REG\_DWORD** seguenti:

**HKEY\_LOCAL\_MACHINE\Software\Microsoft\Command Processor\EnableExtensions\REG\_DWORD**

**HKEY\_CURRENT\_USER\Software\Microsoft\Command Processor\EnableExtensions\REG\_DWORD**

Impostare il valore di **REG\_DWORD** su **0×1** (abilitate) o su **0×0** (disabilitate) nel Registro di sistema mediante Regedit.exe. Le impostazioni specificate dall'utente sono prioritarie rispetto a quelle del computer e le opzioni della riga di comando hanno la precedenza rispetto alle impostazioni del Registro di sistema.

### Attenzione

- Modifiche non corrette al Registro di sistema possono danneggiare gravemente il sistema. Si consiglia di effettuare il backup di tutti i dati più importanti presenti nel computer prima di procedere.

L'abilitazione delle estensioni ai comandi interessa i comandi seguenti:

- **assoc**
- **call**
- **chdir (cd)**
- **color**
- **del (erase)**
- **endlocal**
- **for**
- **ftype**

- **goto**
- **if**
- **mkdir (md)**
- **popd**
- **prompt**
- **pushd**
- **set**
- **setlocal**
- **shift**
- **start** (include le modifiche ai processi di comandi esterni)

Per ulteriori informazioni su questi comandi, fare clic su Argomenti correlati.

- Abilitazione dell'espansione differita delle variabili di ambiente

Se si abilita l'espansione differita delle variabili di ambiente, è possibile utilizzare il punto esclamativo per sostituire il valore di una variabile di ambiente in fase di esecuzione.

- Abilitazione del completamento dei nomi di file e directory

Per impostazione predefinita, il completamento dei nomi di file e directory è disabilitato. Per abilitare o disabilitare il completamento dei nomi di file per un determinato processo del comando **cmd**, utilizzare **/f:{on | off}**. È possibile abilitare o disabilitare il completamento dei nomi di file e directory per tutti i processi del comando **cmd** in un computer o in una sessione di accesso utente impostando i valori di **REG\_DWORD** seguenti:

**HKEY\_LOCAL\_MACHINE\Software\Microsoft\Command Processor\CompletionChar\REG\_DWORD**

**HKEY\_LOCAL\_MACHINE\Software\Microsoft\Command Processor\PathCompletionChar\REG\_DWORD**

**HKEY\_CURRENT\_USER\Software\Microsoft\Command Processor\CompletionChar\REG\_DWORD**

## HKEY\_CURRENT\_USER\Software\Microsoft\Command Processor\PathCompletionChar\REG\_DWORD

Per impostare il valore di **REG\_DWORD**, eseguire Regedit.exe e utilizzare il valore esadecimale di un carattere di controllo per una determinata funzione, ad esempio **0x9** rappresenta TAB, mentre **0x08** BACKSPACE. Le impostazioni specificate dall'utente sono prioritarie rispetto a quelle del computer e le opzioni della riga di comando hanno la precedenza rispetto alle impostazioni del Registro di sistema.

### Attenzione

- Modifiche non corrette al Registro di sistema possono danneggiare gravemente il sistema. Si consiglia di effettuare il backup di tutti i dati più importanti presenti nel computer prima di procedere.

Se si abilita il completamento dei nomi di file e directory mediante **/f:on**, utilizzare CTRL+D per il completamento dei nomi di directory e CTRL+F per quello dei nomi di file. Per disabilitare un determinato carattere di completamento nel Registro di sistema, utilizzare il valore di spazio **[0x20]** perché non è un carattere di controllo valido.

Quando si preme CTRL+D o CTRL+F, il comando **cmd** elabora il completamento dei nomi di file e directory. Queste combinazioni di tasti aggiungono un carattere jolly a *Stringa*, se non è già presente, generano un elenco di percorsi corrispondenti, quindi visualizzano il primo percorso corrispondente. Se nessuno dei percorsi corrisponde, viene emesso un segnale acustico e la visualizzazione del nome non cambia. Per passare da un percorso al successivo, premere ripetutamente CTRL+D o CTRL+F. Per tornare ai percorsi precedenti, premere MAIUSC+CTRL+D o MAIUSC+CTRL+F. Per eliminare l'elenco di percorsi corrispondenti salvato e generarne uno nuovo, modificare *Stringa* e premere CTRL+D o CTRL+F. Se si passa da CTRL+D a CTRL+F e viceversa, l'elenco salvato verrà eliminato e ne verrà generato uno nuovo. L'unica differenza tra le combinazioni di tasti CTRL+D e CTRL+F è che CTRL+D corrisponde solo a nomi di directory, mentre CTRL+F corrisponde sia a nomi di file che di directory. Se si utilizza il completamento dei nomi di file e directory in uno qualsiasi dei comandi di directory incorporati, ovvero **CD**, **MD** o **RD**, il completamento dei nomi di directory verrà eseguito automaticamente.

Il completamento dei nomi di file e directory elabora correttamente i nomi di file contenenti uno spazio o caratteri speciali, a condizione che il percorso corrispondente venga racchiuso tra virgolette.

I seguenti caratteri speciali devono essere racchiusi tra virgolette: & < > [ ] { } ^ \_ = ; ! ' + , ` ~ [spazio]

Se le informazioni fornite contengono spazi, racchiudere il testo fra virgolette, ad esempio "*Nome computer*".

Se si elabora il completamento dei nomi di file e directory all'interno di *Stringa*, qualsiasi parte del *[Percorso]* a destra del cursore verrà eliminata, ovvero a partire dal punto di *Stringa* in corrispondenza del quale è stato elaborato il completamento.

## Convenzioni di formattazione

Formato	Significato
<i>Corsivo</i>	Informazioni che devono essere specificate dall'utente
<b>Grassetto</b>	Elementi che l'utente deve digitare esattamente come indicato
Puntini (...)	Parametro che è possibile ripetere più volte in una riga di comando
Parentesi quadre ([ ])	Elementi facoltativi
Parentesi graffe ({}); possibili scelte separate da una barra verticale ( ). Esempio: {even odd}	Opzioni alternative tra le quali l'utente deve sceglierne una sola
Tipo di carattere Courier	Codice oppure output di programmi

## Vedere anche

### Concetti

[Assoc](#)

[Call](#)

[Endlocal](#)

[For](#)

[Ftype](#)

[Goto](#)

[If](#)

[Prompt](#)

[Imposta](#)

[Shift](#)

[Start](#)

[Informazioni di riferimento per la riga di comando A-Z](#)

[Informazioni introduttive sulla shell dei comandi](#)

### Altre risorse

Chdir  
Color  
Del (erase)  
Mkdir  
Popd  
Pushd  
Setlocal

© 2016 Microsoft

# Call

Aggiornamento: gennaio 2005

Si applica a: Windows Server 2003, Windows Server 2003 R2, Windows Server 2003 with SP1, Windows Server 2003 with SP2

## Call

Richiama un programma batch dall'interno di un altro programma batch senza interrompere l'esecuzione di quest'ultimo. Con il comando **call** è possibile utilizzare etichette come destinazione della chiamata. Il comando **call** non produce alcun effetto dalla riga di comando se utilizzato al di fuori di un file batch o di script.

Sintassi

**call** [[*Unità:*][*Percorso*] *NomeFile* [*ParametriBatch*]] [:*Etichetta* [*Argomenti*]]

Parametri

**[ Unità:][Percorso] NomeFile**

Specifica il percorso e il nome del programma batch che si desidera richiamare. Il parametro *NomeFile* deve avere l'estensione bat o cmd.

**ParametriBatch**

Specifica le informazioni della riga di comando necessarie per il programma batch, inclusi nomi di file, opzioni della riga di comando, parametri batch (ossia valori da %0 a %9) o variabili (ad esempio %baud%).

**: Etichetta**

Specifica l'etichetta a cui passerà il controllo del programma batch. Utilizzando il comando **call** con questo parametro, viene creato un nuovo contesto di file batch e il controllo passa all'istruzione dopo l'etichetta specificata. Dopo aver raggiunto per la prima volta la fine del file batch, ovvero dopo essere passati all'etichetta, il controllo tornerà all'istruzione successiva a **call**. Quando viene raggiunta la fine del file batch per la seconda volta, l'esecuzione dello script batch viene terminata. Per una descrizione dell'estensione **goto :eof**, che consente di uscire da uno script batch, fare clic su Argomenti correlati.

**Argomenti**

Specifica le informazioni della riga di comando che vengono passate alla nuova istanza del programma batch che inizia in corrispondenza di *:Etichetta*, inclusi nomi di file, opzioni della riga di comando, parametri batch (ossia valori da %1 a %9) o variabili (ad esempio %baud%).

**/?**

Visualizza informazioni della Guida al prompt dei comandi.

**Osservazioni**

- Utilizzo di parametri batch

I parametri batch possono contenere informazioni che vengono passate a un programma batch, inclusi nomi di file, opzioni della riga di comando, parametri batch (ossia valori da %0 a %9) o variabili (ad esempio %baud%). Per ulteriori informazioni sui parametri batch, fare clic su Argomenti correlati.

- Utilizzo di pipe e di simboli di reindirizzamento

Non utilizzare pipe e simboli di reindirizzamento con il comando **call**.

- Esecuzione di una chiamata ricorsiva

È possibile creare un programma batch che richiami se stesso, purché venga specificata una condizione di uscita. In caso contrario, si attiverebbe un loop infinito.

- Utilizzo delle estensioni ai comandi

Se le estensioni ai comandi sono abilitate (impostazione predefinita), il comando **call** accetta un'*Etichetta* come destinazione della chiamata. La sintassi corretta è la seguente:

**call:***Argomenti Etichetta*

Per ulteriori informazioni sull'abilitazione e la disabilitazione delle estensioni ai comandi, vedere **Cmd** nell'elenco Argomenti correlati.

## Esempi

Per eseguire un programma denominato Nuovo.bat dall'interno di un altro programma batch, includere in quest'ultimo il seguente comando:

### call nuovo

Se il programma batch da cui si effettua la chiamata accetta due parametri batch e si desidera che tali parametri siano trasferiti a Nuovo.bat, utilizzare il comando seguente nel programma da cui si effettua la chiamata:

**call nuovo %1 %2**

## Convenzioni di formattazione

Formato	Significato
<i>Corsivo</i>	Informazioni che devono essere specificate dall'utente
<b>Grassetto</b>	Elementi che l'utente deve digitare esattamente come indicato
Puntini (...)	Parametro che è possibile ripetere più volte in una riga di comando
Parentesi quadre ([ ])	Elementi facoltativi

Parentesi graffe ({}); possibili scelte separate da una barra verticale ( ). Esempio: {even odd}	Opzioni alternative tra le quali l'utente deve sceglierne una sola
Tipo di carattere Courier	Codice oppure output di programmi

## Vedere anche

### Concetti

[Utilizzo di parametri batch](#)

[Cmd](#)

[Goto](#)

[Utilizzo di parametri batch](#)

[Informazioni di riferimento per la riga di comando A-Z](#)

[Informazioni introduttive sulla shell dei comandi](#)

© 2016 Microsoft

# Date

Aggiornamento: gennaio 2005

Si applica a: Windows Server 2003, Windows Server 2003 R2, Windows Server 2003 with SP1, Windows Server 2003 with SP2

## Date

Visualizza l'impostazione corrente della data di sistema. Se utilizzato senza parametri, il comando **date** visualizza l'impostazione corrente della data di sistema e chiede di digitare una nuova data.

### Sintassi

**date** [MM-GG-AA[AA]] [/t]

### Parametri



***MM - GG - AA[AA]***

Imposta la data specificata, dove *MM* è il mese, *GG* il giorno e *AA[AA]* l'anno.

***/t***

Visualizza la data corrente senza richiedere l'immissione di una nuova data.

***/?***

Visualizza informazioni della Guida al prompt dei comandi.

## Osservazioni

- È necessario separare i valori *MM*, *GG* e *AA[AA]* con punti (.), trattini (-) o barre (/).
- I valori validi per *MM* sono compresi tra 1 e 12.
- I valori validi per *GG* sono compresi tra 1 e 31.
- I valori validi per *AA[AA]* sono compresi tra 80 e 99 o tra 1980 e 2099.
- Windows XP cambia automaticamente il mese e l'anno, quale che sia il numero di giorni del mese (28, 29, 30 o 31).

## Esempi

Per impostare la data sul 3 agosto 2000, digitare:

**08.03.00**

**08-03-00**

**08/03/00**

Per visualizzare la data corrente, digitare:

**date /t**

## Convenzioni di formattazione

Formato	Significato
<i>Corsivo</i>	Informazioni che devono essere specificate dall'utente
<b>Grassetto</b>	Elementi che l'utente deve digitare esattamente come indicato
Puntini (...)	Parametro che è possibile ripetere più volte in una riga di comando
Parentesi quadre ([ ])	Elementi facoltativi
Parentesi graffe ({}); possibili scelte separate da una barra verticale ( ). Esempio: {even odd}	Opzioni alternative tra le quali l'utente deve sceglierne una sola
Tipo di carattere Courier	Codice oppure output di programmi

## Vedere anche

### Concetti

[Ora](#)

[Informazioni di riferimento per la riga di comando A-Z](#)

[Informazioni introduttive sulla shell dei comandi](#)

© 2016 Microsoft

## For

Aggiornamento: gennaio 2005

Si applica a: Windows Server 2003, Windows Server 2003 R2, Windows Server 2003 with SP1, Windows Server 2003 with SP2

## For

Esegue il comando specificato per ciascun file contenuto in un gruppo di file.

Sintassi

**for** {%Variabile | %%Variabile} **in** (Gruppo) **do** Comando [OpzioniRigaComando]

Parametri

**{%Variabile | %%Variabile}**

Obbligatorio. Rappresenta un parametro sostituibile. Utilizzare **%Variabile** per eseguire il comando **for** dal prompt dei comandi. Utilizzare **%%Variabile** per eseguire il comando **for** all'interno di un file batch. Per le variabili viene fatta distinzione tra maiuscole e minuscole ed è necessario rappresentarle con un valore alfabetico, ad esempio %A, %B o %C.

**( Gruppo )**

Obbligatorio. Specifica uno o più file, una o più directory, un intervallo di valori o stringhe di testo che si desidera elaborare mediante il comando specificato. Le parentesi sono obbligatorie.

**Comando**

Obbligatorio. Specifica il comando che si desidera eseguire per ogni file, directory, intervallo di valori o stringa di testo indicata in **(Gruppo)**.

**OpzioniRigaComando**

Specifica le opzioni della riga di comando che si desidera utilizzare con il comando specificato.

**/?**

Visualizza informazioni della Guida al prompt dei comandi.

Osservazioni

- Utilizzo del comando **for**

Il comando **for** può essere utilizzato all'interno di un file batch oppure direttamente dal prompt dei comandi.

- Utilizzo di parametri batch

Con il comando **for** possono essere utilizzati gli attributi seguenti:

- Il comando **for** sostituisce il valore di *%Variabile* o *%%Variabile* con ogni stringa di testo nel *Gruppo* specificato finché il *Comando* non ha elaborato tutti i file.
- Per i nomi *Variabile* del comando **for** viene fatta distinzione tra maiuscole e minuscole. I nomi sono globali e il numero massimo di nomi contemporaneamente attivi è 52.
- Per evitare confusione con i parametri batch compresi tra %0 e %9, è possibile utilizzare qualsiasi carattere per *Variabile*, ad eccezione dei numeri compresi tra 0 e 9. Nei file batch più semplici è sufficiente utilizzare un solo carattere, ad esempio **%%f**.
- Nei file batch più complessi per distinguere le diverse variabili sostituibili, è possibile utilizzare più valori per *Variabile*.

- Specifica di un gruppo di file

Il parametro *Gruppo* può rappresentare un unico gruppo o diversi gruppi di file. Per specificare un gruppo di file è possibile utilizzare i caratteri jolly \* e ?. Di seguito sono riportati esempi di gruppi di file validi.

**(\* .doc)**

**(\* .doc \*.txt \*.me)**

**(gen\*.doc gen\*.rpt feb\*.doc feb\*.rpt)**

**(ar??1991.\* ap??1991.\*)**

Quando si utilizza il comando **for**, il primo valore in *Gruppo* sostituisce *%Variabile* o *%%Variabile* e il comando specificato elabora questo valore. Questa procedura continua finché non sono stati elaborati tutti i file o i gruppi di file corrispondenti al valore di *Gruppo*.

- Utilizzo delle parole chiave **in** e **do**

**In** e **do** non sono parametri, ma parole chiave da utilizzare con il comando **for**. Se si omette una delle due parole chiave, verrà visualizzato un messaggio di errore.

- Utilizzo di forme supplementari del comando **for**

Se le estensioni ai comandi sono abilitate (impostazione predefinita), è possibile utilizzare le forme del comando **for** seguenti:

- Solo directory

Se *Gruppo* contiene caratteri jolly (\* e ?), il *Comando* specificato viene eseguito per ogni directory che corrisponde a *Gruppo* anziché per un gruppo di file in una directory specificata. La sintassi è:

**for/D** {%% | %}Variabile **in** (Gruppo) **do** Comando [OpzioniRigaComando]

- Ricorsiva

Nella struttura delle directory, considerando come directory principale [Unità:]Percorso, esegue l'istruzione **for** in ogni directory della struttura. Se non viene specificata una directory dopo **/R**, il comando verrà eseguito a partire dalla directory corrente. Se *Gruppo* è costituito solo da un punto (.), verrà enumerata solo la struttura della directory. La sintassi è:

**for/R** [[Unità:]Percorso] {%% | %}Variabile **in** (Gruppo) **do** Comando [OpzioniRigaComando]

- Iterazione di un intervallo di valori

Utilizzare una variabile iterativa per impostare il valore di inizio (*NumeroInizio*) e quindi passare attraverso un determinato intervallo di valori fino a quando il valore supera il valore di fine impostato (*NumeroFine*). **/L** eseguirà l'iterazione confrontando il valore di *NumeroInizio* con quello di *NumeroFine*. Se *NumeroInizio* è inferiore a *NumeroFine*, il comando verrà eseguito. Se la variabile iterativa supera il valore di *NumeroFine*, la shell dei comandi uscirà dal loop. È inoltre possibile specificare un valore negativo per *NumeroPassaggio* per passare attraverso un intervallo con valori decrescenti. Ad esempio, (1,1,5) genererà la sequenza 1 2 3 4 5, mentre (5,-1,1) la sequenza (5 4 3 2 1). La sintassi è:

**for/L** {%% | %}Variabile **in** (NumeroInizio,NumeroPassaggio,NumeroFine) **do** Comando [OpzioniRigaComando]

- Iterazione e analisi dei file

Utilizzare l'analisi dei file per elaborare l'output di un comando, stringhe e il contenuto di un file. Utilizzare le variabili iterative per definire il contenuto o le stringhe che si desidera esaminare e utilizzare le diverse opzioni per *ParolechiaveAnalisi* per modificare ulteriormente l'analisi. Utilizzare l'opzione token *ParolechiaveAnalisi* per specificare quali token devono essere passati come variabili iterative. Si noti che, se non si utilizza l'opzione token, **/F** esaminerà unicamente il primo token.

L'analisi dei file consiste nella lettura dell'output, della stringa o del contenuto del file, nella suddivisione in singole righe di testo e quindi nell'analisi di ciascuna riga in zero o più token. Il loop di **for** viene quindi richiamato mediante il valore della variabile iterativa impostato sul token. Per impostazione predefinita, **/F** passa il primo token delimitato da spazi di ogni riga di ogni file. Le righe vuote vengono ignorate. Le diverse sintassi sono:

```
for/F ["ParolechiaveAnalisi"] {%% | %}Variabile in (GruppoNomiFile) do Comando [OpzioniRigaComando]
```

```
for/F ["ParolechiaveAnalisi"] {%% | %}Variabile in ("StringaTesto") do Comando [OpzioniRigaComando]
```

```
for/F ["ParolechiaveAnalisi"] {%% | %}Variabile in ('Comando') do Comando [OpzioniRigaComando]
```

L'argomento *GruppoNomiFile* specifica uno o più nomi di file. Prima di procedere con il file successivo specificato in *GruppoNomiFile*, ogni file viene aperto, letto ed elaborato. Per ignorare il funzionamento predefinito dell'operazione di analisi, specificare "*ParolechiaveAnalisi*". Si tratta di una stringa racchiusa tra virgolette contenente una o più parole chiave per specificare le diverse opzioni dell'analisi.

Se si utilizza l'opzione usebackq, digitare una delle sintassi seguenti:

```
for/F ["usebackqParolechiaveAnalisi"] {%% | %}Variabile in ("GruppoNomiFile") do Comando [OpzioniRigaComando]
```

```
for/F ["usebackqParolechiaveAnalisi"] {%% | %}Variabile in ('StringaTesto') do Comando [OpzioniRigaComando]
```

```
for/F ["usebackqParolechiaveAnalisi"] {%% | %}Variabile in ('Comando') do Comando [OpzioniRigaComando]
```

Nella tabella seguente vengono descritte le parole chiave per l'analisi che è possibile utilizzare per *ParolechiaveAnalisi*.

Parola chiave	Descrizione
eol=c	Specifica un carattere di fine riga (un solo carattere).
skip=N	Specifica il numero di righe da ignorare all'inizio del file.
delims=xxx	Specifica un set di delimitatori. Sostituisce il set di delimitatori predefinito di spazio e tabulazione.
tokens=X,Y,M-N	Specifica i token di ogni riga da passare al comando <b>for</b> per ogni iterazione. Ne consegue che vengono allocati nomi di variabili supplementari. La forma <i>M-N</i> rappresenta un intervallo che specifica i token dalla <i>M</i> fino alla <i>N</i> . Se l'ultimo carattere della stringa <b>tokens=</b> è un asterisco (*), verrà allocata un'ulteriore variabile che riceverà il testo della riga rimasto dopo l'analisi dell'ultimo token.
usebackq	Specifica che è possibile racchiudere tra virgolette i nomi dei file specificati in <i>GruppoNomiFile</i> , che una stringa tra

virgolette inverse verrà eseguita come un comando e che una stringa racchiusa tra virgolette semplici è un comando espresso da una stringa di testo.

#### ○ Sostituzione di variabili

I modificatori di sostituzione per i riferimenti delle variabili del comando **for** sono stati migliorati. Nella tabella seguente vengono descritte le sintassi facoltative, per qualsiasi variabile **I**.

Variabile con modificatore	Descrizione
%~I	Espande %I, il quale rimuove le virgolette di chiusura (").
%~fI	Espande %I in un nome di percorso completo.
%~dI	Espande %I solo in una lettera di unità.
%~pI	Espande %I solo in un percorso.
%~nI	Espande %I solo in un nome di file.
%~xI	Espande %I solo in una estensione di file.
%~sI	Espande il percorso in modo da contenere solo nomi brevi.
%~aI	Espande %I negli attributi del file.
%~tI	Espande %I nella data/ora del file.
%~zI	Espande %I nella dimensione del file.

%~\$PATH:I	Esegue una ricerca nelle directory elencate nella variabile di ambiente PATH ed espande %I nel percorso completo della prima directory individuata. Se il nome della variabile di ambiente non è definito o la ricerca non ha portato all'individuazione del file, questo modificatore si espande nella stringa vuota.
------------	--

Nella tabella seguente vengono descritte le combinazioni di modificatori che è possibile utilizzare per ottenere risultati composti.

Variabile con modificatori abbinati	Descrizione
%~dpI	Espande %I solo in una lettera di unità e in un percorso.
%~nxI	Espande %I solo in un nome di file e una estensione.
%~fsI	Espande %I solo in un percorso completo con nomi di file brevi.
%~dp\$PATH:I	Esegue una ricerca nelle directory elencate nella variabile di ambiente PATH per %I e lo espande nella lettera di unità e nel percorso della prima directory individuata.
%~ftzaI	Espande %I in una riga di output simile a <b>dir</b> .

Negli esempi precedenti %I e PATH possono essere sostituiti con altri valori validi. Un nome di variabile valido per **for** termina la sintassi %~.

L'utilizzo di nomi di variabili espressi con lettere maiuscole come %I può rendere più leggibili le istruzioni ed evitare la confusione con i modificatori, che non fanno distinzione tra maiuscole e minuscole.

- Analisi di una stringa

È possibile utilizzare la logica **for /F** su una stringa trasformando il *GruppoNomiFile* tra parentesi in una stringa racchiusa tra virgolette semplici, ovvero '*GruppoNomiFile*'. *GruppoNomiFile* verrà considerato come un'unica riga di input da un file e quindi sarà analizzato.

- Output dell'analisi

È possibile utilizzare il comando **for /F** per analizzare l'output di un comando trasformando il *GruppoNomiFile* tra parentesi in una stringa racchiusa tra



virgolette inverse. La stringa verrà considerata come una riga di comando, passata a un Cmd.exe figlio, e l'output sarà acquisito in memoria e analizzato come un file.

## Esempi

Per utilizzare il comando **for** in un file batch, utilizzare la sintassi seguente:

**for %% Variabile in ( Gruppo ) do Comando** [OpzioniRigaComando]

Per visualizzare il contenuto di tutti i file nella directory corrente con estensione doc o txt utilizzando la variabile sostituibile %f, digitare:

**for %f in (\*.doc \*.txt) do type %f**

In questo esempio ogni file nella directory corrente con estensione doc o txt verrà sostituito con la variabile %f finché non sarà visualizzato il contenuto di tutti i file. Per utilizzare questo comando in un file batch, sostituire ogni occorrenza di %f con %%f. In caso contrario, la variabile sarà ignorata e verrà visualizzato un messaggio di errore.

Per analizzare un file ignorando le righe di commento, digitare:

**for /F "eol=; tokens=2,3\* delims=," %i in (miofile.txt) do @echo %i %j %k**

Questo comando eseguirà l'analisi di ogni riga contenuta nel file Miofile.txt ignorando le righe che iniziano con un punto e virgola, passando il secondo e il terzo token di ogni riga al comando **FOR**. I token sono delimitati da virgole o spazi. L'istruzione **FOR** indica a %i di individuare il secondo token, a %j di individuare il terzo token e a %k di individuare tutti i token rimanenti. Se i nomi di file indicati contengono spazi, racchiudere il testo fra virgolette, ad esempio "*Nome File*". Per utilizzare le virgolette, è necessario utilizzare la parola chiave **usebackq**. In caso contrario, le virgolette vengono interpretate come se definissero una stringa di testo da analizzare.

La variabile %i è dichiarata esplicitamente nell'istruzione **FOR**. Le variabili %j e %k sono dichiarate esplicitamente mediante **tokens=**. È possibile specificare fino a 26 token nella stringa **tokens=**, a condizione che non si tenti di dichiarare una variabile superiore alla lettera 'z' o 'Z'.

Per analizzare l'output di un comando racchiudendo tra parentesi *GruppoNomiFile*, digitare:

**for /F "usebackq delims==" %i IN ('set') DO @echo %i**

Questo esempio enumera i nomi delle variabili nell'ambiente corrente.

## Convenzioni di formattazione

Formato	Significato
<i>Corsivo</i>	Informazioni che devono essere specificate dall'utente
<b>Grassetto</b>	Elementi che l'utente deve digitare esattamente come indicato
Puntini (...)	Parametro che è possibile ripetere più volte in una riga di comando
Parentesi quadre ([])	Elementi facoltativi
Parentesi graffe ({}); possibili scelte separate da una barra verticale ( ). Esempio: {even odd}	Opzioni alternative tra le quali l'utente deve sceglierne una sola
Tipo di carattere Courier	Codice oppure output di programmi

## Vedere anche

### Concetti

[Cmd](#)

[Type](#)

[Informazioni di riferimento per la riga di comando A-Z](#)

[Informazioni introduttive sulla shell dei comandi](#)

© 2016 Microsoft

## Find

Aggiornamento: gennaio 2005

Si applica a: Windows Server 2003, Windows Server 2003 R2, Windows Server 2003 with SP1, Windows Server 2003 with SP2

# Find

Ricerca in uno o più file una stringa di testo specificata. Al termine della ricerca verranno visualizzate tutte le righe di testo contenenti la stringa specificata.

## Sintassi

**find** [/v] [/c] [/n] [/i] "*Stringa*" [[Unità:][Percorso]NomeFile[...]]

## Parametri

### /v

Visualizza tutte le righe che non contengono la *Stringa* specificata.

### /c

Conta le righe che contengono la *Stringa* specificata e visualizza il totale.

### /n

Fa precedere a ciascuna riga il relativo numero di riga nel file.

### /i

Specifica che la ricerca non deve fare distinzione tra maiuscole e minuscole.

### " *Stringa* "

Obbligatorio. Specifica il gruppo di caratteri da ricercare. Il parametro *Stringa* deve essere racchiuso tra virgolette, ovvero "*Stringa*".

### [Unità:][Percorso] NomeFile

Specifica il percorso e il nome del file in cui si desidera cercare la stringa specificata.

### /?

Visualizza informazioni della Guida al prompt dei comandi.

## Osservazioni

- Specifica di una stringa

Se non si utilizza il parametro **/i**, il comando **find** cercherà esattamente il testo specificato in *Stringa*. Per il comando **find** i caratteri "a" e "A" sono diversi. Se invece si utilizza il parametro **/i**, il comando **find** non fa distinzione tra maiuscole e minuscole e considera "a" e "A" come uno stesso carattere.

Se la stringa che si desidera cercare contiene virgolette ("), sarà necessario utilizzare due caratteri virgolette per ogni carattere virgolette contenuto nella stringa, ovvero "*StringaContenente*"*Virgolette*".

- Utilizzo del comando **find** come filtro

Se si omette un nome di file, **find** si comporterà come un filtro, accettando l'input dall'origine standard, in genere la tastiera, un comando pipe o un file reindirizzato, e visualizzando ogni riga contenente il testo specificato in *Stringa*.

- Ordine della sintassi del comando

È possibile digitare i parametri e le opzioni del comando **find** in qualsiasi ordine.

- Utilizzo dei caratteri jolly

Non è possibile utilizzare i caratteri jolly \* e ? nei nomi di file o nelle estensioni specificate con il comando **find**. Per cercare una stringa in un set di file specificati con i caratteri jolly, utilizzare il comando **find** in un comando **for**.

- Utilizzo di **/v** o **/n** con **/c**

Se i parametri **/c** e **/v** vengono specificati sulla stessa riga di comando, verrà visualizzato il numero delle righe che non contengono la stringa specificata. Se i parametri **/c** e **/n** vengono specificati sulla stessa riga di comando, **find** ignora **/n**.

- Utilizzo del comando **find** con i ritorni a capo

Il comando **find** non riconosce i ritorni a capo. Quando viene utilizzato per la ricerca di testo in file contenenti ritorni a capo, è necessario limitare la stringa di ricerca al testo compreso tra due ritorni a capo, ovvero a una stringa non interrotta da un ritorno a capo. Il comando **find** non rileva ad esempio una corrispondenza per la stringa "spartito musicale" se tra la parola "spartito" e la parola "musicale" è incluso un ritorno a capo.

## Esempi

Per visualizzare tutte le righe del file Matite.ad contenenti la stringa "temperamatite", digitare:

**find "temperamatite" matite.ad**

Per trovare una stringa contenente testo racchiuso tra virgolette, è innanzitutto necessario racchiudere l'intera stringa tra virgolette e quindi utilizzare due

caratteri virgolette per ogni carattere virgolette contenuto nella stringa. Per trovare "Gli scienziati classificarono i documenti come "materiale di discussione". Non si tratta di una relazione finale." nel file Relazione.doc, digitare:

**find "Gli scienziati classificarono i documenti come ""materiale di discussione"". Non si tratta di una relazione finale." relazione.doc**

Per cercare un set di file, utilizzare il comando **find** insieme al comando **for**. Per cercare nella directory corrente tutti i file con estensione bat e che contengono la stringa "PROMPT," digitare:

**for %f in (\*.bat) do find "PROMPT" %f**

Per cercare e visualizzare i nomi dei file sull'unità C del disco rigido che contengono la stringa "CPU", utilizzare il simbolo pipe (|) per indirizzare il risultato di un comando **dir** a **find** digitando:

**dir c:\ /s /b | find "CPU"**

Poiché con il comando **find** è necessario rispettare la distinzione tra maiuscole e minuscole e il comando **dir** visualizza l'output con caratteri maiuscoli, sarà necessario digitare la stringa "CPU" con caratteri maiuscoli oppure utilizzare l'opzione **/i** del comando **find**.

## Convenzioni di formattazione

Formato	Significato
<i>Corsivo</i>	Informazioni che devono essere specificate dall'utente
<b>Grassetto</b>	Elementi che l'utente deve digitare esattamente come indicato
Puntini (...)	Parametro che è possibile ripetere più volte in una riga di comando
Parentesi quadre ([])	Elementi facoltativi
Parentesi graffe ({}); possibili scelte separate da una barra verticale ( ). Esempio: {even odd}	Opzioni alternative tra le quali l'utente deve sceglierne una sola

Tipo di carattere Courier

Codice oppure output di programmi

## Vedere anche

### Concetti

[Utilizzo dei filtri](#)

[Utilizzo degli operatori di reindirizzamento dei comandi](#)

[Informazioni di riferimento per la riga di comando A-Z](#)

[Informazioni introduttive sulla shell dei comandi](#)

© 2016 Microsoft

# Fc

Aggiornamento: gennaio 2005

Si applica a: Windows Server 2003, Windows Server 2003 R2, Windows Server 2003 with SP1, Windows Server 2003 with SP2

# Fc

Confronta due file e visualizza le eventuali differenze.

Sintassi

**fc** [/a] [/b] [/c] [/l] [/lbN] [/n] [/t] [/u] [/w] [/NNNN] [Unità1:][Percorso1]NomeFile1 [Unità2:][Percorso2]NomeFile2

Parametri

**/a**

Abbrevia l'output di un confronto ASCII. Il comando **fc** non visualizza tutte le righe diverse, bensì soltanto la prima e l'ultima riga di ogni gruppo di differenze.

**/b**

Confronta i file in modalità binaria. Il comando **fc** confronta i due file byte per byte, senza tentare una risincronizzazione quando rileva una differenza. Questa è la modalità predefinita per il confronto dei file con le seguenti estensioni: exe, com, sys, obj, lib o bin.

**/c**

Ignora la differenza tra maiuscole e minuscole.

**/l**

Confronta i file in modalità ASCII. Verranno confrontati i due file riga per riga cercando di sincronizzare nuovamente i file dopo aver rilevato una non corrispondenza. Questa è la modalità predefinita per il confronto dei file, tranne quelli con le seguenti estensioni: exe, com, sys, obj, lib o bin.

**/lb N**

Imposta *N* righe per il buffer di riga interno. La lunghezza predefinita per il buffer di riga interno è di 100 righe. Se i file confrontati presentano più di 100 righe consecutive diverse, il comando **fc** annullerà il confronto.

**/n**

Visualizza i numeri di riga durante un confronto in modalità ASCII.

**/t**

Impedisce la conversione dei caratteri di tabulazione in spazi. Per impostazione predefinita, i caratteri di tabulazione vengono considerati come spazi, con interruzioni ogni otto caratteri.

**/u**

Confronta i file come file di testo Unicode.

**/w**

Determina la compressione degli spazi vuoti, ovvero i caratteri di tabulazione e gli spazi, durante il confronto. Se una riga contiene molti spazi o caratteri di tabulazione consecutivi, il parametro **/w** li considererà come un unico spazio. Se il comando **fc** viene utilizzato con l'opzione della riga di comando **/w**, gli spazi vuoti all'inizio e alla fine di una riga verranno ignorati, quindi non confrontati.

**/ NNNN**

Specifica il numero di righe consecutive che devono corrispondere perché il comando **fc** consideri i file risincronizzabili. Se il numero di righe corrispondenti nei file è inferiore a *NNNN*, tali righe saranno visualizzate come differenze. Il valore predefinito è 2.

**[Unità1:][Percorso1]NomeFile1**

Specifica la posizione e il nome del primo file che si desidera confrontare. Il parametro *NomeFile1* è obbligatorio.

**[Unità2:][Percorso2]NomeFile2**

Specifica il percorso e il nome del secondo file che si desidera confrontare. Il parametro *NomeFile2* è obbligatorio.

**/?**

Visualizza informazioni della Guida al prompt dei comandi.

## Osservazioni

- Indicazione delle differenze tra file in un confronto ASCII

Quando si utilizza il comando **fc** per un confronto ASCII, **fc** visualizza le differenze tra i due file nell'ordine seguente:

- Nome del primo file
- Righe di *NomeFile1* che sono diverse rispetto all'altro file
- Prima riga che corrisponde in entrambi i file
- Nome del secondo file
- Righe di *NomeFile2* che sono diverse
- Prima riga che corrisponde

- Utilizzo del parametro **/b** per confronti binari

Il parametro **/b** visualizza le differenze riscontrate durante il confronto binario nel modo seguente:

XXXXXXXX: YY ZZ

Il valore *XXXXXXXX* rappresenta l'indirizzo esadecimale relativo per la coppia di byte, misurato dall'inizio del file. Gli indirizzi iniziano da 00000000. I



valori esadecimali di YY e ZZ rappresentano i byte non corrispondenti rispettivamente di *NomeFile1* e *NomeFile2*.

- Utilizzo dei caratteri jolly

È possibile utilizzare i caratteri jolly \* e ? in *NomeFile1* e *NomeFile2*. Se si utilizza un carattere jolly in *NomeFile1*, verranno confrontati tutti i file specificati con il file indicato da *NomeFile*. Se si utilizza un carattere jolly in *NomeFile2*, il comando **fc** utilizzerà il valore corrispondente di *NomeFile1*.

- Operazioni con la memoria

Per il confronto di due file ASCII, il comando **fc** utilizza un buffer interno, sufficiente a contenere 100 righe. Se i file occupano una quantità di memoria superiore rispetto a quella del buffer, verranno confrontati solo i dati che è possibile caricare nel buffer. Se non viene riscontrata alcuna corrispondenza nelle parti dei file caricate, l'esecuzione verrà interrotta e verrà visualizzato il messaggio seguente:

```
Resynch failed. Files are too different.
```

Durante il confronto di file binari di dimensioni superiori alla memoria disponibile, il comando **fc** confronta i file in maniera completa, sovrapponendo le parti del file in memoria con le parti successive lette dal disco. L'output è uguale a quello generato da un confronto tra file contenuti interamente in memoria.

## Esempi

Per eseguire un confronto ASCII tra due file di testo, Mensile.rpt e Vendite.rpt, e visualizzare i risultati in formato abbreviato, digitare:

**fc /a mensile.rpt vendite.rpt**

Per eseguire un confronto binario tra due file batch, Guadagni.bat e Ricavi.bat, digitare:

**fc /b guadagni.bat ricavi.bat**

Verrà visualizzato il risultato nel seguente formato:

```
00000002: 72 43 00000004: 65 3A 0000000E: 56 92 00000012: 6D 5C 00000013: 0D 7C 00000014: 0D 0A 00000015: 0A 0D 0000001E
```

Se i file Guadagni.bat e Ricavi.bat sono identici, verrà visualizzato il messaggio seguente:

```
FC: no differences encountered
```

Per confrontare tutti i file con estensione bat nella cartella corrente con il file Nuovo.bat, digitare:

**fc \*.bat nuovo.bat**

Per confrontare il file Nuovo.bat sull'unità C con il file Nuovo.bat sull'unità D, digitare:

**fc c:nuovo.bat d:\*.bat**

Per confrontare ogni file batch nella directory principale dell'unità C con il file omonimo nella directory principale dell'unità D, digitare:

**fc c:\*.bat d:\*.bat**

## Convenzioni di formattazione

Formato	Significato
<i>Corsivo</i>	Informazioni che devono essere specificate dall'utente
<b>Grassetto</b>	Elementi che l'utente deve digitare esattamente come indicato
Puntini (...)	Parametro che è possibile ripetere più volte in una riga di comando
Parentesi quadre ([])	Elementi facoltativi
Parentesi graffe ({}); possibili scelte separate da una barra verticale ( ). Esempio: {even odd}	Opzioni alternative tra le quali l'utente deve sceglierne una sola
Tipo di carattere Courier	Codice oppure output di programmi

## Vedere anche

### Concetti

[Informazioni di riferimento per la riga di comando A-Z](#)

Informazioni introduttive sulla shell dei comandi

© 2016 Microsoft

# Findstr

Aggiornamento: gennaio 2005

Si applica a: Windows Server 2003, Windows Server 2003 R2, Windows Server 2003 with SP1, Windows Server 2003 with SP2

## Findstr

Cerca modelli di testo nei file utilizzando espressioni regolari.

Sintassi

**findstr** [/b] [/e] [/l] [/r] [/s] [/i] [/x] [/v] [/n] [/m] [/o] [/p] [/offline] [/g:File] [/f:File] [/c:String] [/d:ElencoDir] [/a:AttributoColore] [Stringhe] [[Unità:][Percorso] NomeFile [...]]

Parametri

**/b**

Trova la corrispondenza con il modello se si trova all'inizio di una riga.

**/e**

Trova la corrispondenza con il modello se si trova a fine riga.

**/l**

Utilizza le stringhe di ricerca in modo letterale.

**/r**

Utilizza le stringhe di ricerca come espressioni regolari. Se non si utilizza il parametro **/l**, il comando **findstr** interpreta tutti i metacaratteri come espressioni regolari.

**/s**

Cerca file corrispondenti nella directory corrente e in tutte le sottodirectory.

**/i**

Specifica che la ricerca non deve fare distinzione tra maiuscole e minuscole.

**/x**

Stampa le righe che corrispondono esattamente.

**/v**

Stampa soltanto le righe che non contengono una corrispondenza.

**/n**

Stampa il numero della riga prima di ciascuna riga corrispondente.

**/m**

Stampa solo il nome del file, se il file contiene una corrispondenza.

**/o**

Stampa la posizione prima di ciascuna riga corrispondente.

**/p**

Ignora i file contenenti caratteri non stampabili.

**/offline**

Elabora i file con l'attributo non in linea.

**/f: *File***

Legge l'elenco di file dal file specificato.

**/c: *Stringa***

Utilizza il testo specificato come stringa di ricerca letterale.

**/g: *File***

Richiama le stringhe di ricerca dal file specificato.

**/d: *ElencoDir***

Effettua la ricerca tra le directory presenti in un elenco delimitato da virgole.

**/a: *AttributoColore***

Specifica gli attributi del colore mediante due cifre esadecimali.

***Stringhe***

Testo specificato da cercare in *NomeFile*.

**[ *Unità:*][ *Percorso*] *NomeFile* [...]**

Specifica uno o più file da cercare.

**/?**

Visualizza informazioni della Guida al prompt dei comandi.

**Osservazioni**

- Utilizzo di *Stringhe* e [*Unità:*][*Percorso*] *NomeFile* [...]

Tutte le opzioni di **findstr** devono precedere *Stringhe* e *[Unità:][Percorso] NomeFile [...]* nella stringa del comando.

- Utilizzo di espressioni regolari con il comando **findstr**

**Findstr** è in grado di individuare in qualsiasi file o set di file ASCII il testo esattamente corrispondente a quello specificato. È tuttavia possibile che si disponga soltanto di parte dei dati da cercare o che si desideri cercare una quantità di dati più ampia. In casi del genere, il comando **findstr** è in grado di cercare modelli di testo tramite espressioni regolari.

Le espressioni regolari sono una notazione per specificare modelli di testo, in alternativa alle stringhe di caratteri esatte. Questa notazione utilizza caratteri letterali e metacaratteri. Ciascun carattere che non ha un significato speciale nella sintassi dell'espressione regolare è un carattere letterale e corrisponde a un'occorrenza di tale carattere. I numeri e le lettere sono ad esempio caratteri letterali. Un metacarattere è un simbolo con un significato speciale nella sintassi dell'espressione regolare, ad esempio un operatore o un carattere di delimitazione.

Nella tabella seguente vengono descritti i metacaratteri accettati dal comando **findstr**.

Carattere	Valore
.	Carattere jolly: qualsiasi carattere
*	Carattere di ripetizione: zero o più occorrenze del carattere o della classe precedente.
^	Posizione nella riga: inizio riga.
\$	Posizione nella riga: fine riga
[ <i>classe</i> ]	Classe di caratteri: qualsiasi carattere compreso nel set.
[^ <i>classe</i> ]	Classe inversa: qualsiasi carattere non compreso nel set.
[ <i>X-y</i> ]	Intervallo: qualsiasi carattere all'interno dell'intervallo specificato.
\X	Escape: utilizzo letterale del metacarattere X.

\<xyz	Posizione nella parola: inizio parola
xyz\>	Posizione nella parola: fine parola

I caratteri speciali sono ancora più efficaci se utilizzati insieme nella sintassi dell'espressione regolare. Ad esempio, la seguente combinazione del carattere jolly (.) e del carattere di ripetizione (\*) corrisponde a qualsiasi stringa di caratteri:

**.\***

Utilizzare l'espressione seguente come parte di un'espressione più lunga che corrisponde a qualsiasi stringa che inizia con "f" e termina con "are":

**f.\*are**

## Esempi

Inserire uno spazio tra le stringhe di ricerca se l'argomento non contiene il prefisso **/c**. Per cercare "buona" o "notte" nel file x.y, digitare:

**findstr "buona notte" x.y**

Per cercare "buona notte" nel file x.y, digitare:

**findstr /c: "buona notte" x.y**

Per cercare tutte le occorrenze della parola "Windows" con l'iniziale maiuscola nel file Proposta.txt, digitare:

**findstr Windows proposta.txt**

Per cercare tutte le occorrenze della parola "Windows", senza distinzione tra maiuscole e minuscole, in tutti i file della directory corrente e nelle relative sottodirectory, digitare:

**findstr /s /i Windows \*.\***

Per cercare tutte le righe che contengono la parola "PER" preceduta da un numero qualsiasi di spazi, come in un ciclo di un programma, e per includere il numero di ogni riga in cui si trova un'occorrenza, digitare:

**findstr /b /n /c:" \*PER" \*.bas**

Per cercare più elementi nello stesso set di file, è necessario creare un file di testo in cui ogni criterio di ricerca occupa una riga. È inoltre possibile elencare in un

file di testo i file in cui si desidera effettuare la ricerca. Per utilizzare il criterio di ricerca nel file Trovadati.txt, cercare i file elencati in Elencofile.txt, quindi memorizzare i risultati nel file Risultati.out, digitare:

**findstr /g:trovadati.txt /f:elencofile.txt > risultati.out**

Per cercare tutti i file che contengono la parola computer nella directory corrente e in tutte le relative sottodirectory, senza distinzione tra maiuscole e minuscole, ed elencare i file in cui si trova la parola cercata, digitare:

**findstr /s /i /m "<computer\>" \*.\***

Per cercare tutte le parole che iniziano con comp, ad esempio complimento e completo, digitare:

**findstr /s /i /m "<comp.\*" \*.\***

## Convenzioni di formattazione

Formato	Significato
<i>Corsivo</i>	Informazioni che devono essere specificate dall'utente
<b>Grassetto</b>	Elementi che l'utente deve digitare esattamente come indicato
Puntini (...)	Parametro che è possibile ripetere più volte in una riga di comando
Parentesi quadre ([])	Elementi facoltativi
Parentesi graffe ({}); possibili scelte separate da una barra verticale ( ). Esempio: {even odd}	Opzioni alternative tra le quali l'utente deve sceglierne una sola
Tipo di carattere Courier	Codice oppure output di programmi

## Vedere anche



## Concetti

Informazioni di riferimento per la riga di comando A-Z

Informazioni introduttive sulla shell dei comandi

© 2016 Microsoft

# Finger

Aggiornamento: gennaio 2005

Si applica a: Windows Server 2003, Windows Server 2003 R2, Windows Server 2003 with SP1, Windows Server 2003 with SP2

## Finger

Visualizza informazioni su uno o più utenti di un computer remoto specificato, in genere un computer che esegue UNIX, che sta eseguendo il servizio Finger o daemon. Il computer remoto specifica il formato e l'output per la visualizzazione delle informazioni sull'utente. Se utilizzato senza parametri, il comando **finger** visualizza informazioni della Guida in linea.

Sintassi

**finger** [-l] [Utente] [@Host] [...]

Parametri

**-l**

Visualizza le informazioni sull'utente in formato lungo.

### ***Utente***

Specifica l'utente su cui si richiedono informazioni. Se si omette il parametro *Utente*, il comando **finger** visualizzerà informazioni su tutti gli utenti del computer specificato.

### **@ Host**

Specifica il computer remoto che sta eseguendo il servizio Finger nel quale si stanno cercando le informazioni sull'utente. È possibile specificare un nome di computer o un indirizzo IP.

**/?**

Visualizza informazioni della Guida al prompt dei comandi.

## Osservazioni

- È possibile specificare più parametri *Utente@Host*.
- Ai parametri di **finger** è necessario anteporre un trattino (-) invece di una barra (/).
- Questo comando è disponibile solo se il **protocollo Internet (TCP/IP)** è installato come componente nelle proprietà di una scheda di rete in Connessioni di rete.
- Nei sistemi operativi della famiglia Windows Server 2003 non è disponibile un servizio finger.

## Esempi

Per visualizzare informazioni su utente1 nel computer utenti.microsoft.com, digitare:

**finger utente1@utenti.microsoft.com**

Per visualizzare informazioni su tutti gli utenti nel computer utenti.microsoft.com, digitare:

**finger @utenti.microsoft.com**

## Convenzioni di formattazione

Formato	Significato
<i>Corsivo</i>	Informazioni che devono essere specificate dall'utente
<b>Grassetto</b>	Elementi che l'utente deve digitare esattamente come indicato
Puntini (...)	Parametro che è possibile ripetere più volte in una riga di comando
Parentesi quadre ([])	Elementi facoltativi
Parentesi graffe ({}); possibili scelte separate da una barra verticale ( ). Esempio: {even odd}	Opzioni alternative tra le quali l'utente deve sceglierne una sola
Tipo di carattere Courier	Codice oppure output di programmi

## Vedere anche

### Concetti

[Informazioni di riferimento per la riga di comando A-Z](#)

[Informazioni introduttive sulla shell dei comandi](#)

© 2016 Microsoft

## Flattemp

Aggiornamento: gennaio 2005

Si applica a: Windows Server 2003, Windows Server 2003 R2, Windows Server 2003 with SP1, Windows Server 2003 with SP2

## Flattemp

Abilita o disabilita le cartelle temporanee condivise.

Sintassi

**flattemp** {/query | /enable | /disable}

Parametri

### /query

Richiede l'impostazione corrente.

### /enable

Abilita le cartelle temporanee condivise.

### /disable

Disabilita le cartelle temporanee condivise.

### /?

Visualizza informazioni della Guida al prompt dei comandi.

Osservazioni

- Dopo la creazione di una directory temporanea univoca per ogni utente, utilizzare **flattemp/enable** per abilitare le directory temporanee condivise.
- Il metodo predefinito per creare le cartelle temporanee per più utenti (in genere indicate dalle variabili d'ambiente TEMP e TMP) consiste nella creazione di sottocartelle nella cartella \Temp, utilizzando l'*ID*accesso come nome della sottocartella. Ad esempio, se la variabile d'ambiente TEMP indica C:\Temp, la cartella temporanea assegnata all'*ID*accesso dell'utente 4 è C:\Temp\4. Utilizzando **flattemp**, è possibile indicare direttamente la cartella \Temp ed evitare che si formino le sottocartelle. Questa soluzione è utile quando si desidera che le cartelle temporanee degli utenti siano contenute nelle home directory, all'interno di un'unità locale di un server terminal o di un'unità di rete condivisa. È necessario utilizzare questo comando solo quando ogni utente ha la propria cartella temporanea separata. Per informazioni sull'utilizzo di cartelle temporanee per ogni sessione, fare clic su Argomenti correlati.

- Per eseguire il comando **flattemp** è necessario disporre di credenziali di amministratore.
- È possibile che si verifichino errori dell'applicazione se la directory temporanea di un utente si trova in un'unità di rete. Questo accade quando le unità di rete condivise diventano temporaneamente inaccessibili in rete. Poiché i file temporanei dell'applicazione sono inaccessibili o non sincronizzati, la risposta che si ottiene è analoga a quella che si riceve in caso di arresto del disco. Si sconsiglia lo spostamento della directory temporanea su un'unità di rete. L'impostazione predefinita prevede di mantenere le directory temporanee sul disco rigido locale. In caso di funzionamento inatteso o di errori di danneggiamento del disco con alcune applicazioni, stabilizzare la rete o spostare le directory temporanee riportandole sul disco rigido locale.
- Se si disabilita l'utilizzo di cartelle temporanee separate per ogni sessione, le impostazioni di **flattemp** verranno ignorate. Questa opzione viene impostata in Configurazione Servizi terminal.

## Esempi

Per visualizzare l'impostazione corrente per le cartelle temporanee condivise, digitare:

**flattemp /query**

Per abilitare le cartelle temporanee condivise, digitare:

**flattemp /enable**

Per disabilitare le cartelle temporanee condivise, digitare:

**flattemp /disable**

## Convenzioni di formattazione

Formato	Significato
<i>Corsivo</i>	Informazioni che devono essere specificate dall'utente
<b>Grassetto</b>	Elementi che l'utente deve digitare esattamente come indicato
Puntini (...)	Parametro che è possibile ripetere più volte in una riga di

	comando
Parentesi quadre ([ ])	Elementi facoltativi
Parentesi graffe ({}); possibili scelte separate da una barra verticale ( ). Esempio: {even odd}	Opzioni alternative tra le quali l'utente deve sceglierne una sola
Tipo di carattere Courier	Codice oppure output di programmi

## Vedere anche

### Concetti

[Informazioni di riferimento per la riga di comando A-Z](#)

[Informazioni introduttive sulla shell dei comandi](#)

[Comandi di Servizi terminal](#)

[Utilizzare cartelle temporanee distinte per ciascuna sessione](#)

© 2016 Microsoft

# Forfiles

Aggiornamento: gennaio 2005

Si applica a: Windows Server 2003, Windows Server 2003 R2, Windows Server 2003 with SP1, Windows Server 2003 with SP2

## Forfiles

Seleziona file in una cartella o in una struttura per l'elaborazione batch.

Sintassi

**forfiles** [/p *Percorso*] [/m *MascheraRicerca*] [/s] [/c *Comando*] [/d[*{+ | -}*] [{*MM/GG/AAAA | GG*}]

## Parametri

### **/p Percorso**

Specifica il *Percorso* sul quale avviare la ricerca. La cartella predefinita corrisponde alla directory di lavoro corrente, specificata da un punto (.).

### **/m MascheraRicerca**

Esegue la ricerca di file in base a *MascheraRicerca*. La *MascheraRicerca* predefinita è \*.\*.

### **/s**

Attiva la ricerca nelle sottodirectory.

### **/c Comando**

Esegue il *Comando* specificato su ogni file. Le stringhe di comando contenenti spazi devono essere racchiuse tra virgolette. Il *Comando* predefinito è **"cmd /c echo @file"**.

### **/d[+ | -] [{MM/GG/AAAA | GG}]**

Seleziona i file con data superiore o uguale (+) oppure inferiore o uguale (-) alla data specificata, dove *MM/GG/AAAA* è la data specificata e *GG* è la data corrente meno *GG* giorni. Se non si specifica + o -, verrà utilizzato +. L'intervallo valido per *GG* è compreso tra 0 e 32768.

### **/?**

Visualizza informazioni della Guida al prompt dei comandi.

## Osservazioni

- **Forfiles** è comunemente utilizzato nei file batch.
- **Forfiles /s** è simile a **dir /s**.
- Nella tabella seguente sono elencate le variabili che è possibile utilizzare nella stringa di comando **/c Comando**.

Variabile	Descrizione
-----------	-------------

@file	Nome file
@fname	Nome file senza estensione
@ext	Estensione del file
@path	Percorso completo del file
@relpath	Percorso relativo del file
@isdir	Valuta se un tipo di file è una directory (TRUE) oppure no (FALSE)
@fsize	Dimensione del file in byte
@fdate	Data ultima modifica del file
@ftime	Ora ultima modifica del file

- Con **forfiles** è possibile eseguire un comando su più file oppure passare argomenti a tali file. È ad esempio possibile eseguire il comando TYPE su tutti i file con estensione txt in una struttura oppure eseguire ogni file batch (\*.bat) sull'unità C:\ con il nome "Mioinput.txt" come primo argomento.
- Con **forfiles** è possibile effettuare le operazioni seguenti:
  - Selezionare file in base a una data assoluta o relativa utilizzando il parametro **/d**.
  - Creare una struttura di file di archivio mediante variabili quali @fsize (dimensione file) e @fdate (data file).
  - Differenziare i file dalle directory utilizzando la variabile @isdir.
  - Formattare l'output includendo nella riga di comando caratteri speciali, preceduti e seguiti dal codice esadecimale **0xHH**.
- Il funzionamento del comando **forfiles** è basato sull'implementazione del flag **recurse subdirectories** sugli strumenti progettati per l'elaborazione di un unico file.

## Esempi



Per elencare tutti i file batch sull'unità C, digitare:

```
forfiles /p c:\ /s /m*.bat /c"cmd /c echo @file è un file batch"
```

Per elencare tutte le directory sull'unità C, digitare:

```
forfiles /p c:\ /s /m*. * /c"cmd /c if @isdir==true echo @file è una directory"
```

Per elencare tutti i file con più di 100 giorni sull'unità C, digitare:

```
forfiles /p c:\ /s /m*. * /dt-100 /c"cmd /c echo @file : data >= 100 giorni"
```

Per elencare tutti i file con data precedente il 1° gennaio 1993 sull'unità C e visualizzare per ognuno di essi il messaggio "La data di *nomefile* non è recente.", digitare:

```
forfiles /p c:\ /s /m*. * /dt-01011993 /c"cmd /c echo La data di @file non è recente."
```

Per elencare tutte le estensioni di tutti i file sull'unità C: in formato colonna, digitare:

```
forfiles /p c:\ /s /m*. * /c "cmd /c echo L'estensione di @file è 0x09@ext0x09"
```

Per elencare tutti i file batch sull'unità C, digitare:

```
forfiles /p c:\ /s /m *.bat /c "cmd /c echo @file è un file batch"
```

Per elencare tutte le directory sull'unità C, digitare:

```
forfiles /p c:\ /s /m *. * /c "cmd /c if @isdir==true echo @file è una directory"
```

Per elencare tutti i file con più di 100 giorni sull'unità C, digitare:

```
forfiles /p c:\ /s /m *. * /d t-100 /c "cmd /c echo @file : data >= 100 giorni"
```

Per elencare tutti i file con data precedente il 1° gennaio 1993 sull'unità C e visualizzare per ognuno di essi il messaggio "La data di *nomefile* non è recente.", digitare:

```
forfiles /p c:\ /s /m *. * /d t-01011993 /c "cmd /c echo La data di @file non è recente."
```

Per elencare tutte le estensioni di tutti i file sull'unità C: in formato colonna, digitare:

```
forfiles /p c:\ /s /m*. * /c "cmd /c echo L'estensione di @file è 0x09@ext0x09"
```

## Convenzioni di formattazione

Formato	Significato
<i>Corsivo</i>	Informazioni che devono essere specificate dall'utente
<b>Grassetto</b>	Elementi che l'utente deve digitare esattamente come indicato
Puntini (...)	Parametro che è possibile ripetere più volte in una riga di comando
Parentesi quadre ([])	Elementi facoltativi
Parentesi graffe ({}); possibili scelte separate da una barra verticale ( ). Esempio: {even odd}	Opzioni alternative tra le quali l'utente deve sceglierne una sola
Tipo di carattere Courier	Codice oppure output di programmi

## Vedere anche

### Concetti

[Informazioni di riferimento per la riga di comando A-Z](#)

[Informazioni introduttive sulla shell dei comandi](#)

[Dir](#)

© 2016 Microsoft

## Formato

Aggiornamento: gennaio 2005

Si applica a: Windows Server 2003, Windows Server 2003 R2, Windows Server 2003 with SP1, Windows Server 2003 with SP2

## Format

Formatta il disco nel volume specificato per consentire l'archiviazione dei file di Windows.

Sintassi

**format** *Volume* [/fs:*FileSystem*] [/v:*Etichetta*] [/q] [/a:*DimensioneUnità*] [/c] [/x]

**format** *Volume* [/v:*Etichetta*] [/q] [/f:*Dimensione*]

**format** *Volume* [/v:*Etichetta*] [/q] [/t:*Tracce/n:Settori*]

**format** *Volume* [/v:*Etichetta*] [/q]

**format** *Volume* [/q]

Parametri

### **Volume**

Specifica il punto di montaggio, il nome del volume o la lettera dell'unità (seguita dai due punti) che si desidera formattare. Se non viene specificata alcuna delle opzioni della riga di comando seguenti, verrà utilizzato il tipo di volume per determinare il formato predefinito per il disco.

### **/fs: *FileSystem***

Specifica il file system da utilizzare: FAT (Tabella di allocazione file, File Allocation Table), FAT32 o NTFS. Per i dischi floppy è possibile utilizzare solo il file system FAT.

### **/v: *Etichetta***

Specifica l'etichetta di volume. Se l'opzione della riga di comando **/v** viene omessa oppure utilizzata senza specificare un'etichetta di volume, verrà chiesta l'assegnazione di un'etichetta di volume al termine della formattazione. Utilizzare la sintassi **/v:** per evitare la richiesta di un'etichetta di volume. Se si formattano più dischi con un solo comando **format**, la stessa etichetta verrà assegnata a tutti i dischi. Per ulteriori informazioni sulle etichette di volume del disco, fare clic su Argomenti correlati.

### **/a: *DimensioneUnità***

Specifica la dimensione delle unità di allocazione da utilizzare con i volumi FAT, FAT32 o NTFS. Se il parametro *DimensioneUnità* viene omissso, il valore verrà impostato sulla base della dimensione del volume. Nella tabella seguente vengono elencati i valori di *DimensioneUnità*.

Valore	Descrizione
<b>512</b>	Crea 512 byte per cluster.
<b>1024</b>	Crea 1024 byte per cluster.
<b>2048</b>	Crea 2048 byte per cluster.
<b>4096</b>	Crea 4096 byte per cluster.
<b>8192</b>	Crea 8192 byte per cluster.
<b>16K</b>	Crea 16 KB per cluster.
<b>32K</b>	Crea 32 KB per cluster.
<b>64K</b>	Crea 64 KB per cluster.

### **/q**

Esegue una formattazione veloce. Elimina la tabella di assegnazione dei file e la directory principale di un volume precedentemente formattato, ma non esegue la scansione settore per settore per verificare la presenza di aree danneggiate. Utilizzare l'opzione della riga di comando **/q** solo per formattare volumi già formattati e privi di settori danneggiati.

### **/f: Dimensione**

Specifica la dimensione del disco floppy da formattare. Se possibile, utilizzare questa opzione della riga di comando anziché le opzioni **/t** e **/n**. In Windows vengono accettati i valori seguenti come dimensione:

**1440** o **1440k** o **1440kb** o **1.44** o **1.44m** o **1.44mb**

Disco floppy da 3,5 pollici, a doppia faccia, a densità quadrupla e con una capacità pari a 1,44 MB

**/t: Tracce**

Specifica il numero di tracce presenti sul disco. Se possibile, utilizzare l'opzione della riga di comando **/f** in alternativa a questa opzione. Se si utilizza l'opzione della riga di comando **/t**, è necessario utilizzare anche l'opzione **/n**. Queste due opzioni insieme rappresentano un metodo alternativo per specificare la dimensione del disco che si sta formattando. Non è possibile utilizzare l'opzione della riga di comando **/f** con l'opzione **/t**.

**/n: Settori**

Specifica il numero di settori per traccia. Se possibile, utilizzare l'opzione della riga di comando **/f** in alternativa a questa opzione. Se si utilizza l'opzione della riga di comando **/n**, è necessario utilizzare anche l'opzione **/t**. Queste due opzioni insieme rappresentano un metodo alternativo per specificare la dimensione del disco che si sta formattando. Non è possibile utilizzare l'opzione della riga di comando **/f** con l'opzione **/n**.

**/c**

Solo NTFS. Imposta la compressione predefinita dei file che verranno creati sul nuovo volume.

**/x**

Se necessario, smonta il volume prima della formattazione. Tutti gli handle del volume aperti non saranno più validi.

**/?**

Visualizza informazioni della Guida al prompt dei comandi.

## Osservazioni

- Utilizzo di **format** dalla console di ripristino di emergenza

Il comando **format** con diversi parametri è disponibile nella console di ripristino di emergenza.

- Credenziali amministrative

Per formattare il disco rigido è necessario essere membri del gruppo Administrators.

- Utilizzo del comando **format**

Il comando **format** crea una nuova directory principale e un nuovo file system del disco. Con questo comando è inoltre possibile verificare l'eventuale presenza di aree danneggiate e cancellare tutti i dati del disco. Affinché possa essere utilizzato un nuovo disco, è necessario innanzitutto formattarlo mediante questo comando.

- Digitazione di un'etichetta di volume

Dopo aver formattato un disco floppy, il comando **format** visualizza il messaggio seguente:

```
Volume label (11 characters, ENTER for none)?
```

L'etichetta di volume può contenere un massimo di 11 caratteri, inclusi gli spazi. Se non si desidera assegnare al disco alcuna etichetta di volume, premere INVIO. Per informazioni sulle etichette di volume, fare clic su Argomenti correlati.

- Formattazione del disco rigido

Quando si utilizza il comando **format** per formattare un disco rigido, prima di eseguire la formattazione, verrà visualizzato un messaggio simile al seguente:

```
AVVERTENZA, TUTTI I DATI SUL DISCO FISSO UNITÀ x: ANDRANNO PERSI Procedere con la formattazione (S/N)? _
```

Per formattare il disco rigido, premere S. Se non si desidera formattarlo, premere N. Per formattare il disco rigido è necessario disporre dei diritti di amministratore.

- Dimensione unità

Il file system FAT limita il numero di cluster a non più di 65526. Il file system FAT32 limita il numero di cluster da un minimo di 65527 a un massimo di 4177917.

La compressione NTFS non supporta unità di allocazione inferiori a 4096.

- Messaggi del comando **format**

Dopo la formattazione, **format** visualizza messaggi relativi allo spazio totale su disco, alle aree contrassegnate come danneggiate e allo spazio disponibile per i file.

- Formattazione veloce

È possibile effettuare una formattazione veloce mediante l'opzione della riga di comando **/q**. Utilizzare questa opzione solo se non sono presenti settori danneggiati sul disco rigido.

- Utilizzo del comando **format** con un'unità riassegnata o di rete

Non utilizzare il comando **format** su un'unità preparata utilizzando il comando **subst**. La formattazione di dischi in rete non è possibile.

- Codici di uscita del comando **format**

La tabella seguente include un elenco dei codici di uscita e una breve descrizione del relativo significato.

Codice di uscita	Descrizione
0	La formattazione è stata eseguita correttamente.
1	Sono stati forniti parametri errati.
4	Si è verificato un errore irreversibile (qualsiasi situazione di errore diversa da quella dei codici 0, 1 o 5).
5	È stato premuto N in risposta al prompt "Continuare con la formattazione (S/N)?" per interrompere l'operazione.

I codici di uscita possono essere controllati utilizzando la condizione **errorlevel** con il comando batch **if**.

## Esempi

Per formattare un nuovo disco floppy sull'unità A utilizzando la dimensione predefinita, digitare:

**format a:**

Per eseguire una formattazione veloce di un disco floppy già formattato in precedenza presente sull'unità A, digitare:

**format a: /q**

Per formattare un disco floppy sull'unità A e assegnargli l'etichetta di volume "DATI", digitare:

**format a: /v:DATI**

## Convenzioni di formattazione

Formato	Significato
<i>Corsivo</i>	Informazioni che devono essere specificate dall'utente
<b>Grassetto</b>	Elementi che l'utente deve digitare esattamente come indicato
Puntini (...)	Parametro che è possibile ripetere più volte in una riga di comando
Parentesi quadre ([])	Elementi facoltativi
Parentesi graffe ({}); possibili scelte separate da una barra verticale ( ). Esempio: {even odd}	Opzioni alternative tra le quali l'utente deve sceglierne una sola
Tipo di carattere Courier	Codice oppure output di programmi

## Vedere anche

### Concetti

[Informazioni introduttive sulla shell dei comandi](#)

[Dir](#)

[If](#)

[Label](#)

[Vol](#)

[Subst](#)

[Informazioni di riferimento per la riga di comando A-Z](#)



# Fsutil:

Aggiornamento: gennaio 2005

Si applica a: Windows Server 2003, Windows Server 2003 R2, Windows Server 2003 with SP1, Windows Server 2003 with SP2

## Fsutil

**Fsutil** è un'utilità da riga di comando che consente di eseguire diverse operazioni relative ai file system FAT e NTFS, ad esempio la gestione dei punti di analisi e dei file di tipo sparse oppure lo smontaggio di un volume. In considerazione delle caratteristiche avanzate di **fsutil**, l'utilizzo è consigliabile solo agli utenti esperti con una conoscenza approfondita di Windows XP o dei sistemi operativi della famiglia Windows Server 2003. Per utilizzare **fsutil**, è inoltre necessario essere connessi come amministratore o come membro del gruppo Administrators.

Nella tabella seguente vengono descritti i sottocomandi **fsutil**.

Comando	Descrizione
<b>Fsutil:</b> <b>behavior</b>	Esegue query, modifica, abilita o disabilita le impostazioni relative alla generazione di nomi di file lunghi 8 caratteri seguiti da un punto e da un'estensione di 3 caratteri, all'utilizzo di caratteri estesi nei nomi di file di questo tipo sui volumi NTFS, all'aggiornamento del timestamp dell'ultimo accesso ai volumi NTFS, alla frequenza di registrazione degli eventi delle quote nel registro eventi di sistema, ai livelli di cache interna della memoria di riserva di paging e non di paging NTFS e alla quantità di spazio su disco riservata alla zona Tabella file master (MTF, Master File Table).
<b>Fsutil:</b> <b>dirty</b>	Esegue una query per verificare se è impostato il dirty bit di un volume. Imposta il dirty bit di un volume. Quando si imposta il dirty bit di un volume, <b>autochk</b> controlla automaticamente il volume per individuare eventuali errori al successivo riavvio del computer.
<b>Fsutil:</b> <b>file</b>	Utilizzato dal personale di supporto tecnico, questo comando cerca un file in base al nome utente (se le quote disco sono abilitate), esegue una query per individuare gli intervalli allocati per un file, imposta il nome breve e la lunghezza dei dati validi di un file oppure imposta su zero i dati di un file.

<b>Fsutil: fsinfo</b>	Utilizzato dal personale di supporto tecnico, questo comando visualizza tutte le unità, esegue query per conoscere il tipo di unità, informazioni sul volume, informazioni sul volume specifiche su NTFS e le statistiche sul file system.
<b>Fsutil: hardlink</b>	Crea un collegamento fisso. Un collegamento fisso è una voce di directory per un file. Si può partire dal presupposto che esista almeno un collegamento fisso per ogni file. Nei volumi NTFS, poiché i file possono avere più collegamenti fissi, un singolo file può essere presente in diverse directory, oppure nella stessa directory con diversi nomi. Poiché tutti i collegamenti fanno riferimento allo stesso file, i programmi possono utilizzare qualsiasi collegamento per aprire il file e modificarlo. I file vengono eliminati dal file system solo dopo l'eliminazione di tutti i collegamenti corrispondenti. Dopo aver creato un collegamento fisso, i programmi possono utilizzarlo come qualsiasi altro nome di file.
<b>Fsutil: objectid</b>	Utilizzato dal personale di supporto tecnico, questo comando gestisce gli identificatori di oggetto utilizzati da Windows XP e dai sistemi operativi della famiglia Windows Server 2003 per tenere traccia di oggetti quali file e directory.
<b>Fsutil: quota</b>	Gestisce le quote disco nei volumi NTFS per offrire un controllo più accurato dell'archiviazione in rete. Le quote disco vengono distribuite per volume e consentono di assegnare i limiti di archiviazione hardware e software per utente.
<b>Fsutil: reparsepoint</b>	Utilizzato dal personale di supporto tecnico, questo comando esegue query o elimina i punti di analisi, ovvero oggetti del file system NTFS ai quali è associato un attributo definibile contenente dati controllati dall'utente e utilizzati per estendere le funzionalità del sottosistema di input/output (I/O). I punti di analisi vengono utilizzati per i punti di giunzione della directory e per i punti di montaggio del volume. Sono inoltre utilizzati dai driver di filtro del file system per contrassegnare particolari file come speciali per il driver specifico.
<b>Fsutil: sparse</b>	Gestisce i file di tipo sparse. Un file di tipo sparse è un file contenente una o più aree di dati non allocati. Per i programmi, tali aree non allocate sono aree contenenti byte con il valore zero, ma non viene in effetti utilizzato spazio su disco per la rappresentazione dei byte zero. In altre parole vengono allocati tutti i dati significativi o diversi da zero, mentre i dati non significativi (lunghe stringhe di dati composte da zero) non vengono allocati. Durante la lettura di un file di tipo sparse, i dati allocati vengono restituiti nella forma memorizzata mentre i dati non allocati vengono restituiti per impostazione predefinita come zero, in conformità con la specifica per i requisiti di protezione C2. Il supporto per i file di tipo sparse consente di deallocare dati da qualsiasi area del file.
<b>Fsutil: usn</b>	Utilizzato dal personale di supporto tecnico, questo comando gestisce il diario delle modifiche del numero di sequenza di aggiornamento (USN), nel quale vengono registrate in modo permanente tutte le modifiche apportate ai file nel volume.
<b>Fsutil: volume</b>	Gestisce un volume, lo smonta o esegue una query per conoscere la quantità di spazio disponibile su un disco.

## Convenzioni di formattazione

Formato	Significato
<i>Corsivo</i>	Informazioni che devono essere specificate dall'utente
<b>Grassetto</b>	Elementi che l'utente deve digitare esattamente come indicato
Puntini (...)	Parametro che è possibile ripetere più volte in una riga di comando
Parentesi quadre ([ ])	Elementi facoltativi
Parentesi graffe ({}); possibili scelte separate da una barra verticale ( ). Esempio: {even odd}	Opzioni alternative tra le quali l'utente deve sceglierne una sola
Tipo di carattere Courier	Codice oppure output di programmi

## Vedere anche

### Concetti

[Fsutil: behavior](#)

[Fsutil: dirty](#)

[Fsutil: file](#)

[Fsutil: fsinfo](#)

[Fsutil: hardlink](#)

[Fsutil: objectid](#)

[Fsutil: quota](#)

[Fsutil: reparsepoint](#)

[Fsutil: sparse](#)

[Fsutil: usn](#)

[Fsutil: volume](#)

[Informazioni di riferimento per la riga di comando A-Z](#)

[Informazioni introduttive sulla shell dei comandi](#)

© 2016 Microsoft

# Ftp

Aggiornamento: gennaio 2005

Si applica a: Windows Server 2003, Windows Server 2003 R2, Windows Server 2003 with SP1, Windows Server 2003 with SP2

## Ftp

Trasferisce file da un computer all'altro eseguendo un servizio FTP (File Transfer Protocol) quale Internet Information Services. Il comando **ftp** può essere utilizzato in modo interattivo o in modalità batch elaborando file di testo in formato ASCII.

Sintassi

**ftp** [-v] [-d] [-i] [-n] [-g] [-s:*NomeFile*] [-a] [-w:*DimensioneFinestra*] [-A] [*Host*]

Parametri

**-v**

Elimina la visualizzazione delle risposte del server FTP.

**-d**

Attiva il debug, visualizzando tutti i comandi passati fra il client FTP e il server FTP.

**-i**

Disabilita le richieste di conferma interattive durante il trasferimento di più file.

**-n**

Elimina l'accesso automatico alla connessione iniziale.

**-g**

Disattiva il glob dei nomi di file. **Glob** consente di utilizzare l'asterisco (\*) e il punto interrogativo (?) come caratteri jolly nei percorsi e nei nomi di file locali. Per ulteriori informazioni, fare clic su Argomenti correlati.

**-s: NomeFile**

Specifica un file di testo contenente i comandi di **ftp**. Questi comandi vengono eseguiti automaticamente dopo l'avvio di **ftp**. Non è possibile inserire spazi in questo parametro. Utilizzare questo parametro al posto del carattere di reindirizzamento (<).

**-a**

Specifica che è possibile utilizzare qualsiasi interfaccia locale durante il binding della connessione dati FTP.

**-w: DimensioneFinestra**

Specifica la dimensione del buffer di trasferimento. La dimensione predefinita è pari a 4096 byte.

**-A**

Accede al server FTP come anonimo.

**Host**

Specifica il nome del computer, l'indirizzo IP o IPv6 del server FTP al quale si desidera stabilire la connessione. Se specificato, il nome o l'indirizzo dell'host deve essere l'ultimo parametro della riga.

**/?**

Visualizza informazioni della Guida al prompt dei comandi.

## Osservazioni

- Ai parametri di **ftp** è necessario anteporre un trattino (-) invece di una barra (/).
- I parametri della riga di comando di **ftp** fanno distinzione tra maiuscole e minuscole.
- Questo comando è disponibile solo se il **protocollo Internet (TCP/IP)** è installato come componente nelle proprietà di una scheda di rete in Connessioni di rete.
- **Ftp** può essere utilizzato in modo interattivo. Dopo l'avvio, **ftp** crea un sottoambiente nel quale è possibile utilizzare i comandi di **ftp**. Per tornare al prompt dei comandi, è sufficiente digitare il comando **quit**. L'esecuzione del sottoambiente **ftp** è indicata dal prompt dei comandi **ftp >**.
- Per ulteriori informazioni sui sottocomandi **ftp**, fare clic su Argomenti correlati.
- Ftp supporta l'utilizzo di IPv6 quando è installato il protocollo IPv6. Per ulteriori informazioni, fare clic su Argomenti correlati.

## Esempi

Per accedere al server FTP denominato ftp.example.microsoft.com, digitare:

**ftp ftp.example.microsoft.com**

Per accedere come utente anonimo al server FTP denominato ftp.example.microsoft.com, digitare:

**ftp -A ftp.example.microsoft.com**

Per accedere al server FTP denominato ftp.example.microsoft.com ed eseguire i comandi di **ftp** contenuti nel file Risinc.txt, digitare:

**ftp -s:risinc.txt ftp.example.microsoft.com**

## Convenzioni di formattazione

Formato	Significato
<i>Corsivo</i>	Informazioni che devono essere specificate dall'utente
<b>Grassetto</b>	Elementi che l'utente deve digitare esattamente come indicato
Puntini (...)	Parametro che è possibile ripetere più volte in una riga di comando
Parentesi quadre ([])	Elementi facoltativi
Parentesi graffe ({}); possibili scelte separate da una barra verticale ( ). Esempio: {even odd}	Opzioni alternative tra le quali l'utente deve sceglierne una sola
Tipo di carattere Courier	Codice oppure output di programmi

## Vedere anche

### Concetti

[Sottocomandi Ftp](#)

[IPv6 \(IP versione 6\)](#)

[Applicazioni IPv6](#)

[Informazioni di riferimento per la riga di comando A-Z](#)

[Informazioni introduttive sulla shell dei comandi](#)

© 2016 Microsoft

# Sottocomandi Ftp

Aggiornamento: gennaio 2005

Si applica a: Windows Server 2003, Windows Server 2003 R2, Windows Server 2003 with SP1, Windows Server 2003 with SP2

## Sottocomandi Ftp

Per ulteriori informazioni, fare clic su un comando:

- [Ftp: !](#)
- [Ftp: help](#)
- [Ftp: append](#)
- [Ftp: ascii](#)
- [Ftp: bell](#)
- [Ftp: binary](#)
- [Ftp: bye](#)
- [Ftp: cd](#)
- [Ftp: close](#)
- [Ftp: debug](#)
- [Ftp: delete](#)
- [Ftp: dir](#)
- [Ftp: disconnect](#)
- [Ftp: get](#)
- [Ftp: glob](#)
- [Ftp: hash](#)
- [Ftp: help](#)



- [Ftp: lcd](#)
- [Ftp: literal](#)
- [Ftp: ls](#)
- [Ftp: mdelete](#)
- [Ftp: mdir](#)
- [Ftp: mget](#)
- [Ftp: mkdir](#)
- [Ftp: mls](#)
- [Ftp: mput](#)
- [Ftp: open](#)
- [Ftp: prompt](#)
- [Ftp: put](#)
- [Ftp: pwd](#)
- [Ftp: quit](#)
- [Ftp: quote](#)
- [Ftp: recv](#)
- [Ftp: remotehelp](#)
- [Ftp: rename](#)
- [Ftp: rmdir](#)
- [Ftp: send](#)

- [Ftp: status](#)
- [Ftp: trace](#)
- [Ftp: type](#)
- [Ftp: user](#)
- [Ftp: verbose](#)

© 2016 Microsoft

# Ftype

Aggiornamento: gennaio 2005

Si applica a: Windows Server 2003, Windows Server 2003 R2, Windows Server 2003 with SP1, Windows Server 2003 with SP2

## Ftype

Visualizza o modifica i tipi di file utilizzati nelle associazioni delle estensioni di file. Se utilizzato senza parametri, il comando **ftype** visualizza i tipi di file per i quali è definita una stringa di comando open.

Sintassi

**Ftype** [*TipoFile* [= [*StringaComandoOpen*]]]

Parametri

**TipoFile**

Specifica il tipo di file che si desidera visualizzare o modificare.

**StringaComandoOpen**

Specifica il comando open da utilizzare quando vengono aperti file di questo tipo.

**/?**

Visualizza informazioni della Guida al prompt dei comandi.

## Osservazioni

- Nella tabella seguente è descritto il modo in cui **ftype** sostituisce le variabili all'interno di una *StringaComandoOpen*.

Variabile	Valore sostitutivo
%0 o %1	Nome del file che si desidera aprire
%*	Tutti i parametri
%~N	Tutti i parametri rimanenti, a partire dall' <i>N</i> esimo parametro, dove <i>N</i> può essere un numero compreso tra 2 e 9
%2	Il primo parametro (la variabile %3 viene sostituita con il secondo parametro e così via)

## Esempi

Per visualizzare i tipi di file correnti per i quali è definita una stringa di comando open, digitare:

**ftype**

Per visualizzare la stringa di comando open corrente per un determinato tipo di file, digitare:

**ftype** *TipoFile*

Per eliminare la stringa di comando open per un determinato tipo di file, digitare:

**f***type* *TipoFile* =

Digitare:

**ASSOC .pl=PerlScript FTYPE PerlScript=perl.exe %1 %\***

Per richiamare lo script Perl, digitare:

**script.pl 1 2 3**

Per eliminare la necessità di digitare le estensioni, digitare:

**set PATHEXT=.pl;%PATHEXT%**

Per richiamare lo script Perl, digitare:

**script 1 2 3**

## Convenzioni di formattazione

Formato	Significato
<i>Corsivo</i>	Informazioni che devono essere specificate dall'utente
<b>Grassetto</b>	Elementi che l'utente deve digitare esattamente come indicato
Puntini (...)	Parametro che è possibile ripetere più volte in una riga di comando
Parentesi quadre ([ ])	Elementi facoltativi
Parentesi graffe ({}); possibili scelte separate da una barra verticale ( ). Esempio: {even odd}	Opzioni alternative tra le quali l'utente deve sceglierne una sola

Tipo di carattere Courier

Codice oppure output di programmi

## Vedere anche

### Concetti

[Assoc](#)

[Start](#)

[Informazioni di riferimento per la riga di comando A-Z](#)

[Informazioni introduttive sulla shell dei comandi](#)

© 2016 Microsoft

# Shift

Aggiornamento: gennaio 2005

Si applica a: Windows Server 2003, Windows Server 2003 R2, Windows Server 2003 with SP1, Windows Server 2003 with SP2

## Shift

Modifica la posizione dei parametri batch in un file batch.

Sintassi

**shift**

Parametri

nessuno

## Osservazioni

- **Utilizzo dell'opzione della riga di comando `shift` con le estensioni ai comandi** Se le estensioni ai comandi sono attivate (impostazione predefinita), il comando **shift** supporta l'opzione della riga di comando `/n`, che indica di iniziare lo spostamento degli argomenti a partire dall'argomento **N**, dove **N** è un valore compreso tra zero e otto. **SHIFT /2** ad esempio sposterà il valore del parametro %3 in %2, il valore del parametro %4 in %3 e così via, lasciando %0 e %1 invariati.
- **Funzionamento del comando `shift`** Il comando **shift** modifica i valori dei parametri batch compresi tra %0 e %9, copiando ogni parametro in quello che lo precede. In altri termini, il valore del parametro %1 verrà copiato in %0, il valore del parametro %2 verrà copiato in %1 e così via. Questa caratteristica risulta utile nello sviluppo di programmi batch che eseguono le stesse operazioni su un numero qualsiasi di parametri.
- **Utilizzo di più di 10 parametri batch** Il comando **shift** può essere utilizzato anche per la creazione di file batch che accettano più di 10 parametri batch. Se nella riga di comando viene specificato un numero di parametri superiore a 10, quelli che compaiono dopo il decimo (%9) verranno spostati uno alla volta nel parametro %9.
- **Utilizzo di %\* con il comando `shift`** Il comando `shift` non ha alcun effetto sul parametro batch %\*.
- **Recupero dei parametri** Non esiste un comando **shift** inverso. Dopo aver eseguito il comando **shift**, non è più possibile ripristinare il valore originario del primo parametro batch (%0).

## Esempi

Il file batch che segue, `Miacopia.bat`, illustra l'utilizzo del comando **shift** con un numero qualsiasi di parametri batch. `Miacopia.bat` copia un elenco di file in una determinata directory. I parametri batch sono rappresentati dagli argomenti `directory` e `nome file`.

```
@echo off rem MIACOPIA.BAT copia un numero qualsiasi di file rem in una directory. rem Il comando utilizza la sintassi s
```

## Convenzioni di formattazione

Formato	Significato
---------	-------------

Corsivo	Informazioni che devono essere specificate dall'utente
Grassetto	Elementi che l'utente deve digitare esattamente come indicato
Puntini (...)	Parametro che è possibile ripetere più volte in una riga di comando
Parentesi quadre ([])	Elementi facoltativi
Parentesi graffe ({}); possibili scelte separate da una barra verticale ( ). Esempio: {even odd}	Opzioni alternative tra le quali l'utente deve sceglierne una sola
Tipo di carattere Courier	Codice oppure output di programmi

## Vedere anche

### Concetti

[Informazioni di riferimento per la riga di comando](#)

[Informazioni introduttive sulla shell dei comandi](#)

[Utilizzo di parametri batch](#)

[Cmd](#)

© 2016 Microsoft

# Utilizzo di parametri batch

Aggiornamento: gennaio 2005

Si applica a: Windows Server 2003, Windows Server 2003 R2, Windows Server 2003 with SP1, Windows Server 2003 with SP2

## Utilizzo di parametri batch

È possibile utilizzare parametri batch in qualsiasi punto all'interno di un file batch per estrarre informazioni sulle impostazioni dell'ambiente.

Cmd.exe rende disponibili le variabili di espansione dei parametri batch da %0 a %9. Quando si utilizzano parametri batch in un file batch, %0 viene sostituito dal nome del file batch mentre le variabili da %1 a %9 vengono sostituite dagli argomenti corrispondenti digitati nella riga di comando. Per accedere ad argomenti oltre il parametro %9, è necessario utilizzare il comando **shift**. Per ulteriori informazioni sul comando **shift**, vedere [Shift](#). Il parametro batch %\* è un riferimento jolly a tutti gli argomenti, escluso %0, che vengono passati al file batch.

Ad esempio, per copiare il contenuto della Cartella1 nella Cartella2 e fare in modo che %1 venga sostituito dal valore Cartella1 e %2 dal valore Cartella2, digitare quanto segue in un file batch denominato Miobatch.bat:

**xcopy %1\\*. \* %2**

Per eseguire il file, digitare:

**miobatch.bat C:\cartella1 D:\cartella2**

In questo modo si otterrà lo stesso risultato ottenuto inserendo quanto segue nel file batch:

**xcopy C:\cartella1 \\*. \* D:\cartella2**

È inoltre possibile utilizzare i modificatori con i parametri batch. I modificatori utilizzano le informazioni sull'unità e la directory correnti per espandere il parametro batch come nome di file o di directory parziale o completo. Per utilizzare un modificatore, digitare il carattere di percentuale (%) seguito da un carattere tilde (~) e quindi digitare il modificatore appropriato (ovvero, **%~modificatore**).

Nella tabella che segue sono elencati i modificatori che è possibile utilizzare in espansione.

Modificatore	Descrizione
<b>%~1</b>	Espande %1 ed elimina le eventuali virgolette (").
<b>%~f1</b>	Espande %1 in un percorso completo.
<b>%~d1</b>	Espande %1 in una lettera di unità.
<b>%~p1</b>	Espande %1 in un percorso.



<b>%~n1</b>	Espande %1 in un nome di file.
<b>%~x1</b>	.Espande %1 in un'estensione di file
<b>%~s1</b>	Il percorso espanso contiene solo nomi brevi.
<b>%~a1</b>	Espande %1 in attributi di file.
<b>%~t1</b>	Espande %1 nella data e ora del file.
<b>%~z1</b>	Espande %1 nella dimensione del file.
<b>%~\$PATH:1</b>	Esegue la ricerca nelle directory elencate nella variabile di ambiente PATH ed espande %1 nel nome completo della prima occorrenza trovata. Se il nome della variabile di ambiente non è definito o la ricerca non ha portato all'individuazione del file, questo modificatore dà come risultato una stringa vuota.

Nella tabella seguente sono riportate le combinazioni possibili di modificatori e qualificatori che è possibile utilizzare per ottenere risultati composti.

<b>Modificatore</b>	<b>Descrizione</b>
<b>%~dp1</b>	Espande %1 in una lettera di unità e in un percorso.
<b>%~nx1</b>	Espande %1 in un nome di file e in un'estensione.
<b>%~dp\$PATH:1</b>	Esegue la ricerca di %1 nelle directory elencate nella variabile di ambiente PATH ed espande %1 nella lettera di unità e nel percorso della prima occorrenza trovata.
<b>%~ftza1</b>	Espande %1 in una riga di output di tipo <b>dir</b> .

#### **Nota**

- Negli esempi precedenti è possibile sostituire **%1** e **PATH** con altri valori di parametri batch.

Il modificatore %\* è un modificatore univoco che rappresenta tutti gli argomenti passati in un file batch. Non è possibile utilizzarlo in combinazione con il modificatore %~. La sintassi di %~ prevede la conclusione con un valore di argomento valido.

Non è possibile utilizzare i parametri batch nel modo in cui vengono utilizzate le variabili di ambiente. Non è inoltre possibile eseguire ricerche e sostituzioni di valori né esaminare sottostringhe. Tuttavia è possibile assegnare il parametro a una variabile di ambiente e quindi intervenire su quest'ultima.

© 2016 Microsoft

# Utilizzo dei filtri

Aggiornamento: gennaio 2005

Si applica a: Windows Server 2003, Windows Server 2003 R2, Windows Server 2003 with SP1, Windows Server 2003 with SP2

## Utilizzo dei filtri

Utilizzato insieme al carattere di reindirizzamento pipe (|), un filtro è un comando all'interno di un altro comando che legge l'input del comando, lo trasforma e quindi scrive l'output. I comandi filtro consentono di ordinare, visualizzare e selezionare parti dell'output di un comando.

I comandi filtro vengono utilizzati per dividere, riordinare o estrarre delle parti di informazioni trasferite tramite essi. Nella tabella seguente sono descritti i comandi filtro disponibili nei sistemi operativi della famiglia Windows Server 2003.

Comando	Descrizione
<a href="#">More</a>	Visualizza il contenuto di un file o dell'output di un comando in una finestra del prompt dei comandi alla volta.
<a href="#">Find</a>	Cerca i caratteri specificati nei file e nell'output del comando.
<a href="#">Sort</a>	Dispone in ordine alfabetico i file e l'output del comando.

Per inviare l'input da un file a un comando filtro, utilizzare il segno di minore (<). Se si desidera che il comando filtro riceva l'input da un altro comando, utilizzare il carattere pipe (|).

## Utilizzo del comando more

Il comando **more** visualizza il contenuto di un file o dell'output di un comando in una finestra del prompt dei comandi alla volta. Per visualizzare ad esempio il contenuto di un file lungo denominato Elenco.txt in una finestra del prompt dei comandi alla volta, digitare:

**more < elenco.txt**

Verrà visualizzata una finestra del prompt dei comandi, in fondo alla quale apparirà il prompt `-- More --`. Per visualizzare la finestra del prompt dei comandi successiva, premere un tasto qualsiasi ad eccezione di PAUSA. Per interrompere il comando senza visualizzare ulteriori informazioni, premere CTRL+C.

Il comando **more** è utile quando si esegue un comando che restituisce un output molto lungo che deve essere visualizzato in più finestre del prompt dei comandi. Si supponga, ad esempio, di volere visualizzare la struttura di directory del disco rigido. Se sono presenti più directory di quante possano essere visualizzate nella finestra del prompt dei comandi, è possibile utilizzare il comando **tree** seguito da un carattere pipe (|) e dal comando **more**, come indicato nell'esempio seguente:

**tree c:\ | more**

Verrà visualizzata una prima finestra del prompt dei comandi con l'output del comando **tree** seguito dal prompt `-- More --`. La visualizzazione dell'output viene sospesa fino a quando non si preme un tasto, ad eccezione di PAUSA.

## Utilizzo del comando find

Il comando **find** cerca il testo o la stringa specificata nei file. Cmd.exe visualizza ogni riga che corrisponde alla stringa o al testo specificato nella finestra del prompt dei comandi. È possibile utilizzare **find** come comando filtro o come comando standard. Per ulteriori informazioni sull'utilizzo di **find** come comando standard, vedere [Find](#).

Per utilizzare **find** come comando filtro, è necessario specificare il segno di minore (<) e la stringa o il testo che si desidera cercare. Per impostazione predefinita, per le ricerche eseguite con il comando **find** viene applicata la distinzione tra maiuscole e minuscole. Il comando seguente troverà ad esempio le occorrenze di "Ditta Bianchi" nel file Clienti.txt:

```
find "Ditta Bianchi" < clienti.txt
```

L'output non include alcuna occorrenza di "ditta bianchi", ma unicamente le occorrenze di "Ditta Bianchi", con le iniziali maiuscole.

Per salvare l'output del comando **find** anziché visualizzarlo nella finestra del prompt dei comandi, digitare un segno di maggiore (>) e il nome del file in cui archiviare l'output. Il comando seguente troverà ad esempio le occorrenze di "Ditta Bianchi" nel file Clienti.txt e le salverà nel file Nclienti.txt:

```
find "Ditta Bianchi" < clienti.txt > Nclienti.txt
```

## Utilizzo del comando sort

Il comando **sort** dispone in ordine alfabetico il contenuto di un file di testo o l'output di un comando. Ad esempio, il comando seguente ordinerà il contenuto del file denominato Elenco.txt e visualizzerà il risultato nella finestra del prompt dei comandi:

```
sort < elenco.txt
```

In questo esempio il comando **sort** dispone in ordine alfabetico le righe del file Elenco.txt e visualizza i risultati senza modificare il file. Per salvare l'output del comando **sort** anziché visualizzarlo, digitare il segno di maggiore (>) seguito dal nome del file. Ad esempio, il comando seguente disporrà in ordine alfabetico le righe del file Elenco.txt e memorizzerà il risultato nel file Elencalf.txt:

```
sort < elenco.txt > Elencalf.txt
```

Per ordinare l'output di un comando, digitare il comando seguito da un carattere pipe (|) e da **sort**, ovvero *Comando* | **sort**. Ad esempio, il comando seguente disporrà in ordine alfabetico le righe che contengono la stringa "Bianchi", ovvero l'output del comando **find**:

```
find "Bianchi" elenco.txt | sort
```