Windows Embedded Standard I filtri sui dischi e sul registro: Introduzione

***di*** [***Beppe Platania***](http://mvp.microsoft.com/it-it/mvp/Beppe%20Platania-4029281) **- Microsoft eMVP**

Blog: <http://beppeplatania.com/it>

* 1. cid:image002.png@01CE4D59.40AA6520

Riveduto e corretto da: [Gianni Rosa Gallina](http://mvp.microsoft.com/it-it/mvp/Gianni%20Rosa%20Gallina-4034912) **- Microsoft eMVP**

Blog: <http://gianni.rosagallina.com/it>

1. *Aprile 2014*

Questo è l’articolo introduttivo di una serie di articoli focalizzati sui filtri sui dischi e sul registro in ambiente Windows Embedded Standard.

L’esigenza di mettere dei filtri sui dischi, soprattutto su quello di sistema, è stata una delle prime scaturite dall’idea di sviluppare un dispositivo embedded. L’obiettivo iniziale era quello di poter avere una “applicazione” dedicata ad uno scopo preciso e che quindi ogni volta che ripartiva aveva le stesse configurazioni senza sapere cosa era successo nel suo periodo di lavoro precedente.

A metà degli anni ’90 cominciarono i tentativi di “protezione” del disco di sistema con vari artifizi e allo stesso tempo iniziò il dilagare di virus e di altri inconvenienti dovuti ad hacker o semplicemente a problemi di sistema.

La soluzione di mettere, in qualche modo, il disco di sistema come un dispositivo di sola lettura sembrava una buona soluzione, ma quando questo veniva ottenuto in maniera “fisica” (uno switch hardware per impedire fisicamente la scrittura) tutti gli applicativi sviluppati per ambienti “desktop”, non aspettandosi degli errori di scrittura, non erano in grado di gestire questo tipo di errore e tendevano a bloccarsi !!

Ancora più problematica è la situazione dal punto di vista del sistema operativo: scritto per funzionare nativamente su un disco rigido su cui scrive con un’alta frequenza un elevato numero di informazioni sia necessarie a tener traccia delle evoluzioni del sistema, sia a livello storico per poter risalire, a posteriori, ad eventuali problemi verificatisi nel corso dell’utilizzo: file di log, file temporanei, ecc...

Per poter funzionare nativamente senza potersi appoggiare su un disco si sarebbero dovuti riscrivere, o rivedere, pezzi di sistema operativo! In alternativa si sarebbe dovuta trovare una soluzione, esterna al cuore del sistema, per ottenere gli stessi risultati senza creare o modificare pesantemente il sistema stesso.

Parallelamente all’idea “funzionale” di proteggere il sistema si è concretizzata la possibilità di utilizzare memorie flash al posto dei dischi rigidi con vantaggi legati all’eliminazione di parti in movimento, cioè diminuire l’usura e allungare la “vita” del dispositivo di massa. L’utilizzazione di memorie flash, inoltre, è fondamentale per utilizzi in cui il dispositivo è soggetto a forti sollecitazioni fisiche: scossoni e vibrazioni o ambienti particolari come quelli polverosi.

Le memorie flash hanno, però, un numero di scritture limitato (anche se nel tempo si è passati dal milione di riscritture sulla stessa cella a più del doppio, pur rimanendo in costi contenuti!). L’effetto che si ottiene installando un sistema operativo Windows su di una flash senza l’abilitazione di un filtro di scrittura è che nel tempo (anche poche settimane, dipende dalla flash, dal sistema e dagli applicativi) la dimensione della flash diminuisce fino a non permettere più nessuna scrittura e il sistema comincia ad avere problemi.

Se, dopo l’installazione del sistema e degli applicativi non si scrivesse più (o si scrivesse molto poco) sul disco, la “vita” della flash si allungherebbe considerevolmente (decine di anni e non più settimane!).

Microsoft, già dai tempi di Windows NT Embedded, ha ricercato una soluzione semplice ed efficace e questa si è concretizzata, fino a ieri, in due funzionalità distinte: il filtro di scrittura avanzato (EWF = Enhanced Write Filter) e il filtro di scrittura su file (FBWF = File Based Write Filter).

Con il rilascio di Windows Embedded 8 Standard le due funzionalità sono state “integrate” in un “filtro unificato” (UWF = Unified Write Filter) che ha migliorato, sotto vari aspetti, la fruibilità di questa tecnologia.

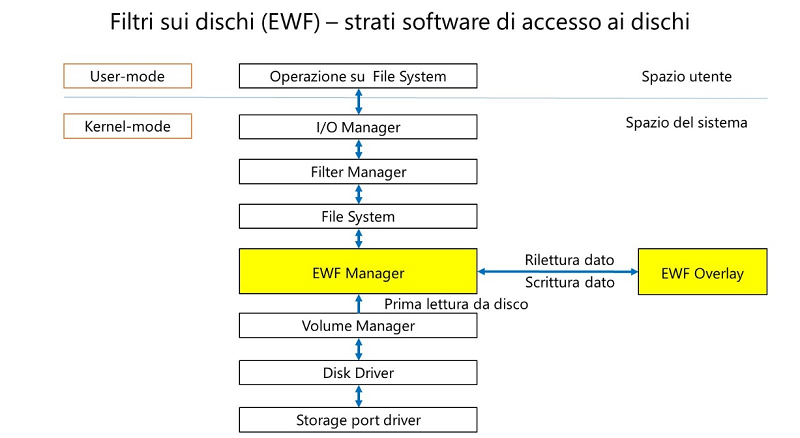
Il meccanismo di funzionamento è quello di mettere un filtro per non aggiornare “fisicamente” il disco di sistema e memorizzare le modifiche che questi subisce durante la sessione di lavoro in un altro posto, tipicamente la RAM oppure un altro disco; quando il sistema riparte, il disco non è stato modificato e ci si ritrova sempre nella stessa situazione iniziale.

L’architettura di Windows permette l’introduzione di questo filtro in più parti dello “stack di scrittura” per cui i filtri principali si differenziano proprio per il posto nello stack dove effettuano il loro compito.

Il filtro di scrittura avanzato (EWF=Enhanced Write Filter) viene applicato ad un livello molto vicino al driver HW di scrittura sul disco, potremmo dire che, nel momento in cui il sistema ha individuato il “settore” di disco da cui dovrebbe leggere, controlla se questo ha il filtro abilitato o meno. In caso positivo controlla se il dato è già stato aggiornato nella sessione corrente, in caso positivo legge il dato dal luogo (Overlay) dove la configurazione ha scelto di salvare i dati, in caso negativo legge “fisicamente” dal disco.

Analogamente per la scrittura: nel momento in cui il sistema ha individuato il “settore” di disco su cui dovrebbe scrivere, controlla se questa ha il filtro abilitato o meno. In caso positivo scrive il dato nel luogo (Overlay) dove la configurazione ha scelto di salvare i dati, in caso negativo scrive “fisicamente” sul disco.

Nella figura seguente riportiamo uno schema grossolano degli strati di sistema per arrivare al “settore” fisico del disco dove si trova l’informazione evidenziando la posizione del filtro di scrittura avanzato (di qui in avanti ***filtro-EWF***).

* 1. 

Per poter spiegare meglio le funzionalità messe a disposizione dai filtri di sistema, partiamo con introdurre il concetto di overlay. Abbiamo detto che il filtro di scrittura protegge il contenuto del disco ridirigendo le operazioni di scrittura in un altro luogo, chiamato overlay. L’overlay è il posto dove vengono memorizzate tutte le modifiche del disco protetto dal filtro, mentre la configurazione del filtro e la “lista” di dove sono state fatte queste modifiche può essere scritta o su una partizione particolare di un disco rigido o in memoria (nel registro).

Un “overlay” può essere visto come se fosse un filtro fotografico posto su un’immagine, il filtro contiene i cambiamenti (es: la variazione di colore con cui risulta proiettata l’immagine), ma l’immagine sotto rimane immutata. Quando si toglie il filtro si rivede l’immagine originale. Seguendo questo concetto quando si esegue la funzione di “commit”, cioè si riporta il contenuto dell’overlay su disco è come se avessi “incollato” il mio filtro all’immagine variandone il contenuto, nel nostro esempio: il colore.

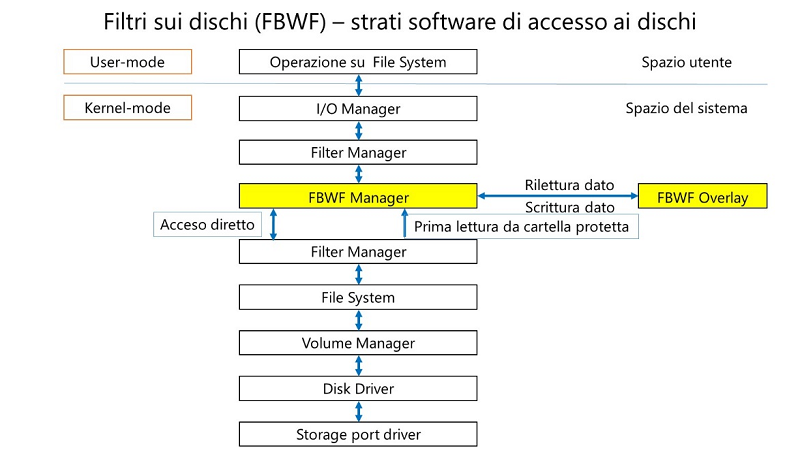
Il filtro di scrittura con memorizzazione su overlay era presente già nella versione embedded di Windows NT, ma non era facilmente gestibile, poi è stato migliorato in Windows XP Embedded, da qui il nome Enhanced. Con i rilasci successivi, nel Future pack 2007 di XP Embedded, al filtro avanzato si è aggiunto il filtro basato su file (FBWF=File Base Write Filter) e il filtro sul registro.

La necessità di avere un filtro a livello di “file system” si è evidenziata con la necessità di avere una parte del disco “sprotetta” perché utilizzata per svariate motivazioni: file di log che non si riescono a spostare su altri volumi, cartelle di dati che chi ha scritto l’applicazione ha scelto di memorizzare nel disco di sistema, ecc…

Inoltre, un’altra richiesta era quella di poter memorizzare in modo permanente delle modifiche fatte a dei file senza dover aggiornare, in un’unica soluzione, l’intero sistema o fare un riavvio del dispositivo.

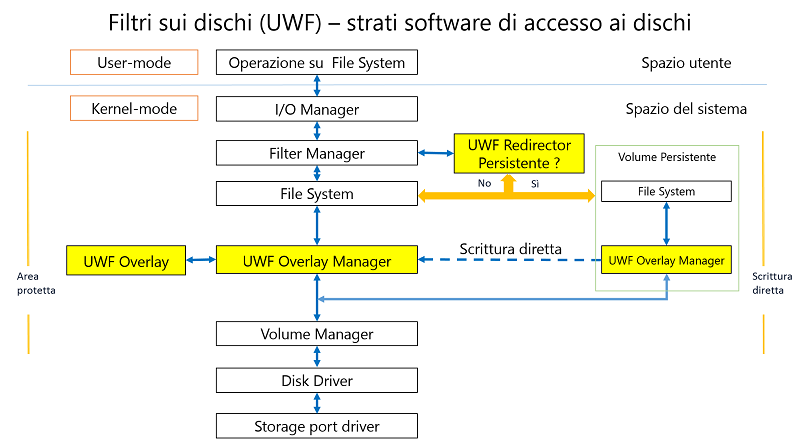
Il filtro basato su file (da qui in avanti ***filtro-FBWF***) viene applicato ad un livello più alto rispetto a quello del ***filtro-EWF***, in un layer dove il sistema ha la capacità di gestire i file. A questo punto, infatti, invece di proteggere o meno l’intera partizione si può scegliere nel dettaglio quali file e/o cartelle lasciare sprotette.

Nella figura seguente riportiamo, come nello schema precedente, gli strati di sistema per arrivare a scrivere fisicamente su disco, ma, come si può notare, il ***filtro-FBWF*** è posizionato più in alto rispetto al ***filtro-EWF***; in un punto dove si hanno tutte le informazioni legate al “file system”.

* 1. 
  2. L’operatività sull’overlay rimane simile: invece di memorizzare dati legati a riferimenti di “settori” del disco come nel ***filtro-EWF*** , si memorizzano più semplicemente i nomi di file e/o cartelle. In questo modo NON c’è più bisogno di creare e memorizzare una lista di ciò che si sta proteggendo o meno, perché si opera utilizzando direttamente quella della configurazione del ***filtro-FBWF***.

Con l’arrivo di Windows Embedded 8 Standard c’è stata un’evoluzione con l’introduzione di un nuovo filtro di scrittura su disco Unified Write Filter (UWF) che da qui in avanti chiameremo ***filtro-UWF***. Questo filtro è stato chiamato così perché mette insieme le proprietà dei due filtri precedenti: ***filtro-EWF*** e ***filtro-FBWF*** che, da questa versione diavanti, vengono sconsigliati. I vantaggi portati da questo filtro sono legati ad una migliore integrazione con il sistema e con alcune sue funzionalità critiche come l’aggiornamento delle liste per l’antiMalware.

Nella figura seguente riportiamo, come negli schemi precedenti, gli strati di sistema per arrivare a scrivere fisicamente su disco. Il ***filtro-UWF***, mettendo insieme le potenzialità dei due filtri precedenti, è più complesso ed agisce a più livelli (ne parleremo approfonditamente in articoli a lui dedicati).

* 1. 

Tutti questi filtri sui dischi hanno numerose interazioni con altre funzionalità embedded. La più interessante è quella che si crea con l’abilità che ha Windows Embedded Standard di avviare il sistema da un dispositivo USB sia che questo sia un disco rigido con un adattatore o che sia un dispositivo flash. In questo secondo caso, oltre al fatto già citato che le flash abbiano un numero limitato di riscritture, bisogna evidenziare anche che la velocità di scrittura su flash è molto più lenta (10-20Mb/sec) di quella in memoria. Utilizzando uno dei filtri sui dischi salvaguardiamo la flash poiché memorizza le modifiche su un overlay (che è in memoria) e velocizziamo il sistema.

## Il filtro sul registro (Registry Filter)

Quando si utilizza un filtro di scrittura su disco, indipendentemente che si usi un ***filtro-EWF***, un ***filtro-FBWF*** o un ***filtro-UWF***, automaticamente viene protetto anche il registro. Il filtro sul registro è stato creato perché tutte le modifiche al registro vengono memorizzate sul disco di sistema e, se questi è protetto da scrittura, verranno perse al prossimo riavvio e questo è un problema per tutte le informazioni che necessitano di essere mantenute anche nella sessione successiva.

Il primo ad avere bisogno di questa funzionalità è stato chi utilizza un accesso ad un “Dominio” o una connessione RDP. In tutti e due i casi, infatti, la controparte server assegna, alla prima connessione, un “token” di licenza da “presentare” nelle connessioni successive. Se questo “token” non viene presentato, il server ritiene di essere di fronte ad una nuova connessione e ne assegna un altro. I “token” però sono legati alle licenze e quindi NON si devono sprecare.

Fino a Windows Embedded Standard 2009 questo filtro era limitato a queste due chiavi, poi si è evidenziato che, se il dispositivo deve utilizzare un applicativo che ha i parametri di configurazione memorizzati sul registro, ad ogni riavvio, perdendo il registro, avrebbe perso anche tutte le modifiche a questi parametri durante la sessione di lavoro.

Se l’applicativo è sotto il nostro controllo, quando lo integriamo in un sistema protetto da scrittura, possiamo spostare i parametri in un file di configurazione posizionato in posto sprotetto. Se però l’applicativo è di terze parti NON siamo in grado di modificarlo e quindi non saremmo in grado di utilizzarlo se non con i parametri di default. Con il filtro sul registro possiamo definire un certo ramo del registro che verrà salvato in modo da non essere perduto al prossimo riavvio.

Nella versione Windows Embedded Standard 2009 sono presenti sia il ***filtro-EWF*** sia il ***filtro-FBWF*** e sia il filtro sul registro con opzioni e capacità simili alla versione Windows Embedded Standard 7. Nell’ultima versione 8 Standard si è aggiunto il filtro unificato UWF (Unified Write Filter) di qui in avanti ***filtro-UWF***, per cui, nel corso dell’approfondimento dei parametri legati ad ogni filtro, della loro configurazione e gestione, evidenzieremo anche le differenze tra le varie versioni di Windows Embedded Standard.

## Considerazioni generali in presenza di filtri di scrittura

Prima di addentrarci nelle caratteristiche di ogni filtro di scrittura, facciamo qualche riflessione su alcune funzionalità del sistema Windows che, in presenza di filtri di scrittura sul disco (in particolare su quello di sistema), vanno neutralizzate o quanto meno riviste:

|  |  |
| --- | --- |
|  |  |
| System Restore | Questa funzionalità permette di memorizzare dei “dati di ripristino” nel caso in cui si voglia (o si debba) tornare indietro dopo un’installazione o un aggiornamento. Nel caso in cui il disco di sistema è protetto da scrittura gli aggiornamenti devono venir organizzati in un modo più complesso e quindi questi “dati di ripristino” potrebbero rivelarsi inutili e quindi eliminare questo servizio potrebbe far migliorare l’efficienza del sistema. |
| Background Defrag | Questa funzionalità permette la deframmentazione dei dischi in background (mentre il sistema sta effettuando altre operazioni) senza disturbare le altre applicazioni che stanno lavorando. Molto apprezzato dagli utenti desktop può creare dei grossi problemi se il disco di sistema è una flash che patirebbe questo utilizzo spasmodico della riscrittura. Nel caso in cui il disco è protetto da scrittura questa deframmentazione avrebbe l’unico risultato di riempire l’overlay di scritture inutili. |
| Prefetch, Superfetch | Queste funzionalità, in modalità diverse, fanno caricare al sistema le DLL delle applicazioni più usate prima che queste vengano richieste in modo da migliorare i tempi di attivazione delle applicazioni stesse. Nel caso in cui il disco di sistema è protetto da scrittura la memorizzazione degli elementi da caricare non sopravvivrebbe da una sessione alla successiva rendendo questa funzionalità inapplicabile. |
| Last Access Time Stamps | Questa funzionalità aggiorna la data e l’ora d’accesso ad ogni file di un disco con formattazione NTFS. Nel caso in cui il disco è protetto da scrittura questa memorizzazione avverrebbe comunque in memoria occupando spazio senza dare nessun vantaggio. Bisogna tener presente che alcuni applicativi si basano su queste informazioni per effettuare o meno delle operazioni sui file. Per gestire questa funzionalità bisogna operare direttamente sul sistema intervenendo sul registro:  **Chiave: HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\FileSystem**  **Nome: NtfsDisableLastAccessUpdate**  **Tipo: REG\_DWORD**  **Valore: 1** |
| Temporary Files Folder | Nel caso in cui il disco è protetto da scrittura il consiglio è di spostare o sopprimere tutte le memorizzazioni temporanee: le cache di appoggio degli applicativi e/o le cartelle TEMP, TMP che normalmente il sistema crea per utilizzarle alla prossima sessione. In questo caso, se il dispositivo avesse oltre a quello di sistema un altro volume non protetto da scrittura, sarebbe meglio spostare lì queste cartelle in modo da utilizzarle al meglio. Ad esempio:  **Chiave: HKEY\_CURRENT\_USER\Software\Microsoft\Windows\Current Version\Explorer\User Shell Folders**  **Nome: Cache**  **Tipo: REG\_EXPAND\_SZ**  **Valore: <percorso di un volume non protetto>**  **-------**  **Chiave: HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders**  **Nome: Cache**  **Tipo: REG\_EXPAND\_SZ**  **Valore: <percorso di un volume non protetto>**  **-------**  **Chiave: HKEY\_CURRENT\_USER\Environment**  **Nome: TEMP**  **Tipo: REG\_SZ**  **Valore: <percorso di un volume non protetto>**  **-------**  **Chiave: HKEY\_CURRENT\_USER\Environment**  **Nome: TMP**  **Tipo: REG\_SZ**  **Valore: <percorso di un volume non protetto>** |
| Pagefile | Il *pagefile* è il file di sistema utilizzato per poter superare i limiti della memoria fisica del dispositivo: quando il sistema lo ritiene opportuno usa questo disco come luogo di appoggio di aree di RAM non utilizzate che vengono poi riprese quando l’evolversi della sessione lo richiede. Nel mondo Windows Embedded Standard la configurazione di default è “disabilitato” nel senso che il sistema NON utilizza il *pagefile*. Se si hanno altri volumi oltre a quello di sistema, il *pagefile* può essere spostato su uno dei volumi sprotetti. In caso contrario è meglio lasciarlo disabilitato. Ad esempio per metterlo sul volume D:  **Chiave: HKEY\_CURRENT\_USER\Software\Microsoft\Windows\Current Version\Explorer\User Shell Folders**  **Nome: Cache**  **Tipo: REG\_EXPAND\_SZ**  **Valore: <percorso di un volume non protetto>** |
| Event Log Location | Nel caso in cui il disco è protetto da scrittura il consiglio è di spostare (ad esempio in rete dove un amministratore di sistema potrà gestirle) o sopprimere anche tutte le segnalazioni di sistema. Se il dispositivo avesse, oltre a quello di sistema, un altro volume non protetto da scrittura, sarebbe meglio spostare lì questo posizionamento dei file di sistema in modo da poterli utilizzare da una sessione alla successiva. Ad esempio spostando le segnalazioni sul volume D:  **Chiave: HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\EventLog\Application\**  **Nome: File**  **Tipo: REG\_EXPAND\_SZ**  **Valore: <percorso di un volume non protetto>\AppEvent.evt**  **-------**  **Chiave: HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\EventLog\Security\**  **Nome: File**  **Tipo: REG\_EXPAND\_SZ**  **Valore: <percorso di un volume non protetto>\SecEvent.evt**  **-------**  **Chiave: HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\EventLog\System\**  **Nome: File**  **Tipo: REG\_EXPAND\_SZ**  **Valore: <percorso di un volume non protetto>\SysEvent.evt** |
| Boot Status Policy | Con questo termine si intende il modo con cui il sistema, al suo avvio, deve gestire gli eventuali errori di chiusura o di incongruenza determinatisi nella sessione precedente. Ad esempio quando il sistema non si è chiuso con uno shutdown normale, ma a seguito un a mancanza di corrente. In questo caso all’avvio (in situazioni da desktop) viene presentata una pagina a caratteri in cui si evidenzia che il sistema NON è stato chiuso correttamente e che quindi “potrebbero” esserci dei problemi. Nel caso embedded ed in particolare quando il disco di sistema è protetto da scrittura, questa pagina oltre ad essere inutile è anche dannoso: un sistema embedded protetto di solito si spegne così proprio perché non ci sono altri modi (basta pensare ad una cassa di un supermercato o a un chiosco informativo) e spesso, dopo tanti sforzi per togliere tutti i brand possibili perché si vuole che il sistema si presenti soltanto con i “colori” dell’OEM che lo ha costruito, questa pagina fa immediatamente riconoscere l’ambiente Windows e, dalla segnalazione inopportuna, erroneamente se ne dedurrebbe un punto di debolezza inesistente! |
| Automatic Adjustment of  Daylight Saving Time | In Italiano suona come “adeguamento automatico all’ora legale”. Il problema nasce dal fatto che, in presenza di un filtro sul registro, quando cambia l’ora legale (sia in un verso che nell’altro) il sistema, dopo aver portato avanti (o indietro) l’ora, memorizza nel registro che questa operazione è stata conclusa. Ma, al successivo riavvio, il registro ritorna senza le modifiche della sessione e quindi, appena il sistema controlla se deve cambiare l’ora lo fa nuovamente!  Per risolvere questo problema ci sono più possibilità:  Se utilizzate un *filtro-UWF* potete aggiungere le seguenti chiavi di registro alla lista delle chiavi da escludere dalla protezione:  **HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Time Zones**  **HKLM\SYSTEM\CurrentControlSet\Control\TimeZoneInformation**  Se utilizzate un *filtro-EWF* o un *filtro-FBWF* potete aggiungere le seguenti chiavi di registro alla lista del *filtro sul registro*:  **HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\TimeZoneInformation\RealTimeIsUniversal**  In ogni caso, un’altra possibilità è quella di immettere nella data/ora del BIOS quella UTC (Coordinated Universal Time) diciamo: “l’ora solare di Greenwich” e di informare il sistema di questa scelta. In questo modo il sistema Windows Standard esegue i calcoli correttamente e, anche se il registro è protetto da scrittura, non ci sono problemi. Per eseguire questa configurazione operate in questo modo:  Disabilitate i filtri di scrittura su disco (in modo da disabilitare anche quelli sul registro);  Controllate di aver configurato correttamente il fuso orario desiderato;  Aggiungete al registro:  **Chiave: HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\TimeZoneInformation\**  **Nome: RealTimeIsUniversal**  **Tipo: DWORD**  **Valore: 1**  Fate ripartire il sistema ed entrate nel BIOS;  Selezionate la data e l’ora UTC. Ad esempio se siete nel fuso orario italiano in estate e sono le 11 del mattino impostate l’ora alle 9 poiché in Italia, durante l’estate, siamo 2 ore avanti rispetto a Greenwich;  Fate ripartire il sistema ed abilitate il filtro sui dischi (ovvero sul registro) |
| Service write protect devices | In Italiano suona come “aggiornamento di sistema su un disco di sistema protetto da scrittura”. Il Problema è dovuto al fatto che, se il disco è pretetto, tutti gli aggiornamenti effettuati durante una sessione verranno persi al prossimo riavvio!! Per gestire correttamente un aggiornamento bisognerà agire in questo modo:  Disabilitate i filtri di scrittura su disco (in modo da disabilitare anche quelli sul registro);  Riavviare il dispositivo;  Eseguire tutti gli aggiornamenti necessari ri-avviando il sistema se e quando necessario;  Abilitate i filtri di scrittura su disco (in modo da abilitare anche quelli sul registro);  Riavviare il dispositivo; |

Di queste funzionalità e di come fare a gestirle mediante i tool di Windows Embedded Standard, ne parleremo nei prossimi articoli che riguarderanno degli approfondimenti per ognuno dei filtri sui dischi.

***di*** [***Beppe Platania***](http://mvp.microsoft.com/it-it/mvp/Beppe%20Platania-4029281) **- Microsoft eMVP**

Blog: <http://beppeplatania.com/it>

Riveduto e corretto da: [Gianni Rosa Gallina](http://mvp.microsoft.com/it-it/mvp/Gianni%20Rosa%20Gallina-4034912) **- Microsoft eMVP**

Blog: <http://gianni.rosagallina.com/it>