Windows Embedded Standard I filtri sui dischi e sul registro: EWF - Enhanced Write Filter

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

#### 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-4034912http:/mvp.microsoft.com/it-it/mvp/Gianni%20Rosa%20Gallina-4034912) - Microsoft eMVP

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

1. *Maggio 2014*

[Nell’articolo precedente](http://msdn.microsoft.com/it-it/library/dn722310.aspx) ci siamo interessati ai filtri di scrittura in generale, in questo ci occuperemo di quello che va sotto il nome di filtro di scrittura avanzato ***filtro-EWF*** (Enhanced Write Filter).

Il filtro di scrittura avanzato (***filtro-EWF***) viene applicato a livello di “settore” (detto anche a livello di volume perché si applica globalmente ad un intero volume) e può essere applicato su una o su più partizioni/volumi. Il ***filtro-EWF***, rendendo praticamente il disco in sola lettura, permette l’avvio del sistema da dispositivi in sola lettura come CD, dischi in LAN, ecc…. In Windows Embedded Standard 7 e 8 Standard quest’ultima funzionalità non è più disponibile, nel senso che questi due sistemi, per operare correttamente, necessitano di un supporto “fisicamente scrivibile”, anche se i filtri limiteranno queste operazioni al minimo indispensabile.

In Windows Embedded Standard 7 e 8 Standard sono previste due “modalità” di configurazione dell’overlay del ***filtro-EWF***: ***RAM*** e ***RAM-Reg***. In entrambe le modalità l’overlay è posizionato in memoria mentre la configurazione del filtro e la “lista” dei settori contenenti i dati elaborati (in pratica: ciò che è da ricercare dall’overlay e non dal disco fisico) è posizionata su:

una partizione del disco per la configurazione ***RAM*** (la partizione inizialmente NON va formattata perché verrà gestita dal ***filtro-EWF*** con una formattazione proprietaria);

nel registry (e quindi in memoria) per la configurazione ***RAM-Reg (default)*.**

La soluzione ***RAM-Reg*** è quella da utilizzare in presenza di Compact Flash (tipicamente NON partizionabili in più volumi e quindi NON utilizzabili con la soluzione ***RAM***) o di altri dispositivi non configurabili come dischi “fissi” (in contrapposizione a “removibili”). La scelta ***RAM-reg*** è di aiuto anche in sede di produzione dei dispositivi quando bisogna “clonare” la build “MASTER” (chiamiamo così la build che è diventata l’immagine finale da distribuire). Non avendo partizioni con una formattazione proprietaria, la duplicazione risulta più semplice.

Per riassumere la funzionalità quindi il ***filtro-EWF*** ha bisogno di:

un posto dove annotarsi la “***lista***” di quali sono i “settori” dove si sarebbe dovuto scrivere il dato.

un ***overlay*** dove scrivere il “dato” che non scriviamo direttamente sul disco fisico e, in tutti e due le modalità (***RAM*** e ***RAM-reg***), l’overlay è in memoria.

Nella versioni precedenti di Windows Embedded: XPe e Standard 2009 il ***filtro-EWF*** prevedeva anche una modalità (***DISK***) in cui l’overlay veniva scritto su una partizione dedicata di un disco, si potevano avere fino a nove overlay di questo tipo e si poteva richiedere la ripartenza del sistema da uno qualunque degli overlay memorizzati. Windows Embedded Standard 7 e 8 Standard non prevedono più questa modalità ***DISK*** per il ***filtro-EWF.***

Tutte le altre opzioni di questa funzionalità, che vedremo più avanti, servono a “modellare” il funzionamento del ***filtro-EWF***.

Prerequisiti di sistema

Nel pianificare l’utilizzo di questa funzionalità è bene controllare le caratteristiche del disco che si vuole proteggere e tutta una serie di prerequisiti di sistema:

***Modalità RAM*** 🡪 durante l’installazione del ***filtro-EWF*** esso stesso crea un nuovo “volume” e quindi, se non trova tutte le condizioni per potersi “annotare” gli indirizzi su disco, la funzionalità NON verrà attivata correttamente. Ecco le situazioni più comuni che portano a questi problemi:

il disco ha già raggiunto il numero massimo di volumi creabili per disco (normalmente 4 per i dischi primari): l’installatore non riesce a creare un nuovo volume e ... fallisce;

non è stato lasciato lo spazio sul disco per creare il nuovo volume: ha bisogno di almeno 32KB di disco non partizionati;

il disco è “dinamico”, il filtro lavora soltanto con dischi denominati “Basic”;

il disco è visto dal sistema come “removibile” e non “fisso”.

Il consiglio è quello di mantenere, in ogni caso, la configurazione più semplice possibile: il disco di sistema nella partizione C: protetta dal ***filtro-EWF*** che NON occupa tutto il disco, permettendo così all’agente di installazione del ***filtro-EWF*** di creare la propria partizione per gli indici. E’ importante ricordare che la tipologia di volume creata dall’installatore del disco NON è una di quelle tipiche del mondo Windows (es: FAT32, NTFS, ecc..) e quindi NON è visibile dagli applicativi standard di sistema. Questo comporta il vantaggio di proteggere i dati da manipolazioni non volute, ma, allo stesso tempo, comporta alcune complicazioni per la duplicazione dei dischi necessaria al momento della “produzione” dei dispositivi di cui parleremo più avanti.

***Modalità RAM-Reg*** 🡪 questa modalità NON necessita della creazione di un nuovo “volume”: la lista degli indirizzi viene memorizzata nel registro.

## Gli elementi del filtro-EWF

La tecnologia del ***filtro-EWF*** utilizza un certo numero di elementi che si occupano, ognuno per il proprio compito, di eseguire tutte le funzionalità che il filtro prevede: dalla gestione dell’abilitazione e disabilitazione all’avvio fino alla possibilità di memorizzare i cambiamenti fino ad un certo momento avvenuti con una chiamata a sistema.

Il package che contiene tutti questi elementi si chiama “***Enhanced Write Filter***”. Nel catalogo è posizionato nel ramo: Features\Lockdown\Compatibility Write Filters\ e contiene gli elementi descritti nella seguente tabella:

|  |  |
| --- | --- |
| Elemento | Descrizione |
| Ewf.sys | E’ il driver di sistema del ***filtro-EWF***. Colui che si occupa di intercettare le richieste di scrittura e lettura dal disco che si sta proteggendo, indirizzando la funzione al disco fisico o all’overlay a seconda dei casi. |
| Ewfcfg.exe | E’ l’eseguibile che viene lanciato al momento dell’installazione degli elementi del ***filtro-EWF*** o durante il loro aggiornamento. L’eseguibile si limita a lanciare la corrispondente DLL (Ewfcfg.dll) che gestisce tutte le funzionalità necessarie. |
| Ewfcfg.dll | I parametri del ***filtro-EWF***, configurati tramite il tool ICE (= Image Configuration Editor), vengono gestiti da questa DLL in vari momenti della vita del sistema: all’installazione, all’aggiornamento o quando si usa Sysprep per preparare il sistema alla duplicazione. |
| Ewfmgr.exe | Ewfmgr.exe (EWF Manager Console Application) è l’applicazione a riga commando che permette di conoscere lo stato del ***filtro-EWF*** e la sua configurazione attuale. Inoltre, offre una serie di comandi per modificare lo stato del ***filtro-EWF*** e la sua configurazione (più avanti vedremo la lista dei comandi in dettaglio). |
| Ewfapi.dll | Ewfapi.dll (Enhanced Write Filter API) è la libreria di interfaccia per poter gestire il ***filtro-EWF*** dall’interno di un applicativo. Tutte le richieste di stato ed i comandi che si possono gestire mediante l’applicativo Ewfmgr.exe, si possono gestire con delle chiamate (API) verso questa libreria. |
| Horm.dat | E’ un file di configurazione che contiene le informazioni del ***filtro-EWF*** e, in particolare dell’HORM (Hibernate Once, Resume Many) per le fasi di avvio del sistema (ne riparleremo approfonditamente più avanti). |

In aggiunta a questi elementi, già dal SP1 di Standard 7 è stato aggiunto un package (Enhanced Write Filter Management Tool) per la gestione a run-time, con un’interfaccia grafica, del ***filtro-EWF*** (ne parleremo più avanti).

Al momento dell’installazione del ***filtro-EWF***, in caso di modalità ***RAM***, viene creata (dalla DLL EWFcfg.dll) una piccola partizione (<64Kb) chiamata ***Volume-EWF***. In questa partizione vengono memorizzate tutte le informazioni di configurazione del ***filtro-EWF***: numero dei volumi da proteggere, dimensioni, ecc.... E’ da notare che verrà creato un singolo ***Volume-EWF*** anche se viene protetto più di un volume. Nei dati memorizzati nel ***Volume-EWF*** ci saranno anche i comandi di riavvio che abilitano o disabilitano il ***filtro-EWF*** stesso.

In caso di modalità ***RAM-Reg*** tutte le informazioni di cui abbiamo parlato vengono memorizzate nel registro.

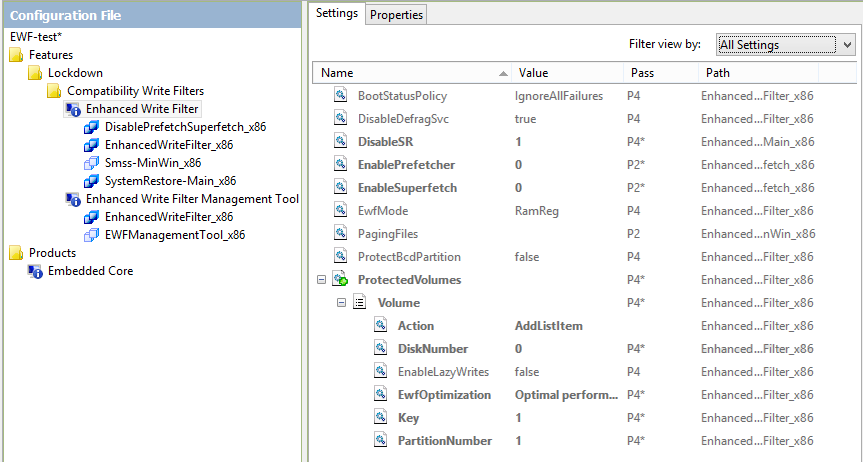
Nei parametri di configurazione di questa funzionalità

|  |  |
| --- | --- |
| Nome | Descrizione |
| DisableSR | Determina se disabilitare o meno la memorizzazione dei dati “di ripristino” del sistema.  Nel nostro caso specifico, spesso conviene evitare che il sistema scriva dei dati “di ripristino” sul disco che, protetto dal ***filtro-EWF***, li scriverebbe di fatto in memoria e al riavvio, se non salvati con un comando specifico, li perderebbe.  0; Abilita la memorizzazione dei dati “di ripristino”;  1; Disabilita la memorizzazione dei dati “di ripristino” (**default**). |
| DisableDefragSvc | Determina se disabilitare o meno il servizio di deframmentazione disco.  Questo servizio, utile se si sta lavorando con un disco rigido, diventa inutile e dannoso se si sta lavorando con una flash che ha già internamente un sistema di organizzazione delle scritture ottimizzato per la propria natura.  **True**; Disabilita il Servizio (**default**);  **False**; Abilita il Servizio. |
| EnablePrefetcher | La funzionalità di “Prefetch” fa in modo che il sistema cominci a caricare dei dati degli applicativi più utilizzati dall’utente nella sessione prima ancora che questi ne chieda nuovamente il caricamento. Tutto questo velocizza i tempi di risposta percepiti dall’utente nel momento in cui richiede i successivi caricamenti.  0; Disabilita la funzionalità (**default**);  1; Abilita la funzionalità. |
| EnableSuperfetch | La funzionalità di “Superfetch” è una miglioria alla funzionalità di “Prefetch” estendendola oltre la sessione: il sistema memorizza gli applicativi più utilizzati dall’utente e comincia a precaricarli nella prossima sessione prima ancora che l’utente ne chieda il caricamento. Tutto questo velocizza i tempi di risposta percepiti dall’utente nel momento in cui richiede applicativi che aveva utilizzato nelle sessioni precedenti. In caso specifico di utilizzo del ***filtro-EWF***, questa memorizzazione verrebbe persa al riavvio e quindi in generale la funzionalità è meglio lasciarla disabilitata in modo da non affaticare il sistema con azioni inutili.  0; Disabilita la funzionalità (**default**);  1; Abilita la funzionalità. |
| PagingFiles | Determina il nome ed il posizionamento del file di ***paging*** del sistema. Il file di ***paging*** è il luogo dove il sistema si appoggia quando è in carenza di RAM. Nel nostro caso, dove stiamo utilizzando il ***filtro-EWF***, utilizzare un ***paging*** potrebbe portare ad un controsenso: il sistema è in carenza di RAM e tenta di scrivere su un ***paging*** file che è su un volume protetto e che quindi scriverebbe su un overlay che è esso stesso in RAM!!  Ci sono casi in cui, invece di disabilitare la funzione, conviene spostare il ***paging*** file su un altro volume. Ad esempio: potrei avere il sistema su un disco a ***tecnologia flash*** protetto dal ***filtro-EWF*** e utilizzare un ***paging*** file posizionato su un altro disco a ***tecnologia classica***.  Come **default** la funzionalità è disabilitata ed il valore di questo parametro è una stringa vuota. |
| BootStatusPolicy | Determina, in caso di errori che si dovessero presentare all’avvio o al riavvio del sistema, quali debbano essere visualizzati. Poiché stiamo operando proprio per poter spegnere il dispositivo direttamente interrompendo l’alimentazione, è meglio evitare che, alla ripartenza del sistema, venga visualizzato un qualsiasi messaggio d’errore.  DisplayAllFailures Visualizza tutti gli errori;  IgnoreAllFailures Non visualizzare errori (**default**);  IgnoreBootFailures Non visualizzare gli erori di avvio del sistema;  IgnoreShutdownFailures Non visualizzare gli errori di chiusura del sistema. |
| EwfMode | Determina la modalità con cui si vuole utilizzare il ***filtro-EWF*** :  **RAM**; vedi descrizioni fatte in precedenza;  **RAM-Reg**; vedi descrizioni fatte in precedenza (**default**). |
| ProtectBcdPartition | Determina se il ***filtro-EWF*** debba proteggere o meno la partizione di avvio BCD (= Boot Configuration Data). In questo caso la scelta è molto legata all’applicazione specifica: ad esempio  **True**; il ***filtro-EWF*** protegge anche la partizione di boot **False**; il ***filtro-EWF*** non la protegge (**default**); |
| ProtectedVolumes | Elenca la lista dei volumi (partizioni) che devono essere protette dal ***filtro-EWF***. Se il volume non è presente in questa lista, non potrà essere protetto a “run-time”.  Questo è un parametro “complesso” che prevede, per ogni volume inserito, un’ulteriore lista di parametri elencati nella tabellina che segue. |

Lista dei parametri per ogni volume configurato nel ***filtro-EWF***.

|  |  |
| --- | --- |
| Nome | Descrizione |
| Action | Determina come deve essere inizializzato il volume.  **AddListItem** La configurazione del volume viene aggiunta all’immagine (**default**);  **Modify** La configurazione del volume viene aggiornata;  **RemoteListItem** La configurazione del volume viene rimossa. |
| Key | E’ un numero positivo, che parte da 1, per identificare il volume che si vuole proteggere. Ogni Volume **deve** avere una ***Key*** univoca. |
| EnableLazyWrites | Determina se i dati nell’overlay devono essere scritti in “background” o al momento della richiesta.  **True** Abilita la scrittura in background;  **False** Scrive al momento della richiesta (**default**); |
| DiskNumber | E’ un numero positivo, **che parte da 0**, per identificare il disco secondo l’**ARC part** (come viene presentato dall’utility ***DISKPART***). |
| PartitionNumber | E’ un numero positivo, **che parte da 1**, per identificare il volume secondo l’**ARC part** (come viene presentato dall’utility ***DISKPART***). |
| EwfOptimization | E’ un parametro per configurare le “performance” del ***filtro-EWF***.  Optimal performance  Le “migliori prestazioni”: il ***filtro-EWF*** copia l’intero settore del disco nell’overlay occupando la maggior quantità di memoria, ma ottimizzando i tempi d’accesso (**default**);  Use less overlay space  Usa minor spazio possibile dell’overlay: il ***filtro-EWF*** controlla se i dati da scrivere sono già gli stessi contenuti su disco e copia nell’overlay soltanto quelli cambiati;  Use less overlay space and less writes  Usa minor spazio possibile dell’overlay e minimizza le scritture: il ***filtro-EWF*** controlla se i dati da scrivere sono già gli stessi contenuti su disco e se i dati nell’overlay sono uguali a quelli che dovrebbe scrivere e, in questo caso, non li riscrive. |

Nella figura seguente vediamo come si presenta la configurazione dei parametri del ***filtro-ewf*** nel configuratore ICE.



La configurazione del filtro-EWF a riga comando

Ewfmgr.exe è l’applicativo a livello console che permette di interagire con i servizi e le configurazioni del ***filtro-EWF.*** L’applicazione per funzionare correttamente ha bisogno dei privilegi di amministrazione e si trova nelle cartelle di sistema (%windowsdir%\system32), quindi è accessibile da qualsiasi cartella.

Quasi tutti i comandi del ***filtro-EWF*** hanno bisogno di un riavvio per diventare operativi.

La sintassi:

ewfmgr [<volume-name>\*](optional) [-all] [-commit] [-commitanddisable [-live]]

[-disable] [-enable] [-nocmd] [-persist="<persistent data>"]

Vediamo i vari parametri in dettaglio:

|  |  |
| --- | --- |
| Nome | Descrizione |
| All | Richiede l’applicazione dello specifico comando a tutti i volumi protetti. |
| Commit | Richiede l’aggiornamento del volume fisico con i dati contenuti nell’overlay. Il comando **Commit** può essere associato a quello di **Disable** per disabilitare il ***filtro-EWF***. L’overlay verrà scritto al prossimo avvio del sistema che quindi lo startup può esserne rallentato. |
| CommitandDisable | Richiede l’aggiornamento del volume fisico con i dati contenuti nell’overlay. L’overlay verrà scritto al prossimo avvio del sistema che quindi lo startup può esserne rallentato. Con questo comando si può richiedere l’opzione **Live** sia se si sta utilizzando la modalità RAM che si usi quella RAM-Reg. Con l’opzione Live l’aggiornamento viene effettuato immediatamente mentre la disabilitazione del filtro avverrà al prossimo avvio. |
| Disable | Richiede la disabilitazione del filtro per uno specifico volume.  NOTA: Nella modalità RAM-Reg i dati non sopravvivono al riavvio in quanto sono memorizzati nel registry e quindi, se si vuole memorizzare il contenuto dell’overlay sul disco, l’utilizzo dei comandi **CommitandDisable** **-live** diventa obbligatorio. |
| Enable | Richiede l’abilitazione del filtro per uno specifico volume. |
| NoCmd | Annulla tutti i comandi pendenti. |
| Persist | Specifica un campo a 64-Byte che viene mantenuto attraverso gli overlay per lo stesso volume. |
| ActivateHorm | Richiede l’attivazione della funzionalità di HORM al prossimo avvio.  In **Standard 8** si consiglia di utilizzare l’HORM con il ***filtro-UWF*** piuttosto che con il ***filtro-EWF.*** |
| DeactivateHorm | Richiede la disattivazione della funzionalità di HORM al prossimo avvio.  In **Standard 8** si consiglia di utilizzare l’HORM con il ***filtro-UWF*** piuttosto che con il ***filtro-EWF.*** |

Tutti questi parametri di configurazione e tutti i comandi che abbiamo scorso possono essere richiesti anche in forma programmatica: il tool mette a disposizione una libreria di comandi C++ (EWF API) ed una serie di esempi di programmazione che permettono di utilizzare il ***filtro-EWF*** dall’interno di un’applicazione con tutti i vantaggi che questo comporta:

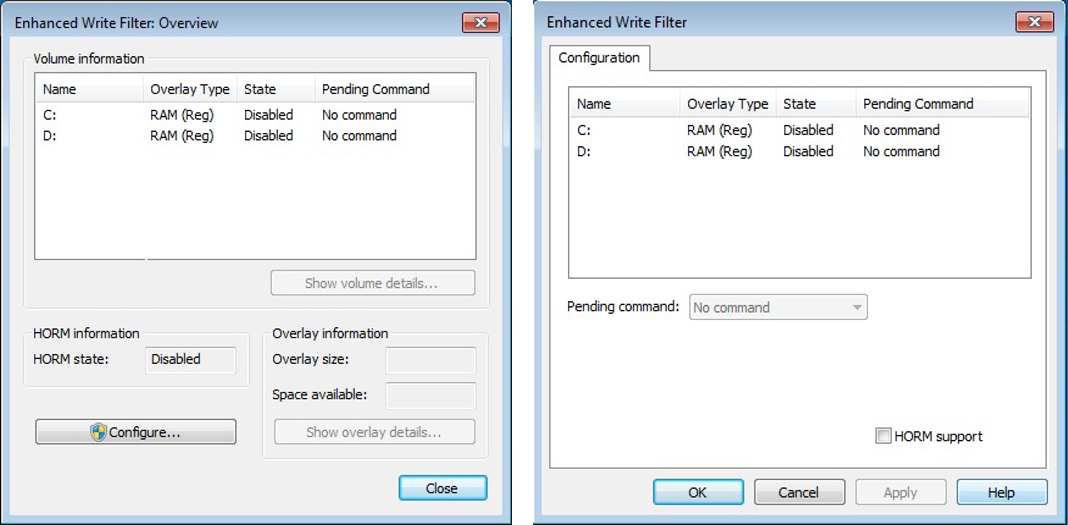
NON dover chiedere all’utente di effettuare operazioni da riga comando sul ***filtro-EWF***;

Effettuare le richieste nel momento opportuno (dal punto di vista programmatico) ad esempio per richiedere l’ibernazione quando tutti gli applicativi coinvolti hanno finito le loro inizializzazioni.

Poter “tracciare” su dei file di log le operazioni fatte in modo da poter controllare a posteriori ed evitare eventuali contestazione.

## La configurazione del filtro-EWF ad interfaccia grafica

EwfMgmt.exe è l’applicativo ad interfaccia grafica che permette di interagire con i servizi e le configurazioni del ***filtro-EWF.*** L’applicazione, una volta inserita nella build, viene automaticamente inserita nella taskbar della shell standard. Per funzionare correttamente ha bisogno dei privilegi di amministrazione e si trova nelle cartelle di sistema (%windowsdir%\system32), quindi è accessibile da qualsiasi cartella. In molti casi, dopo aver determinato in laboratorio le configurazioni del ***filtro-EWF*** che si vogliono utilizzare e come automatizzare un minimo queste scelte, è meglio cancellare questo applicativo che, per la sua semplicità d’uso, potrebbe portare l’utente finale a compiere operazioni inopportune.

* 1. 

## La gestione del filtro-EWF via interfaccia programmabile (API), Power-Shell e WMI.

Il ***filtro-EWF*** fornisce, insieme al toolkit, tutta la struttura per un’interfaccia programmabile in grado di configurare e gestire tutte la caratteristiche del filtro. Durante l’installazione del toolkit i file di supporto per utilizzare queste **API** (Application Program Interface) vengono posizionati (come default) in queste cartelle:

C:\Program Files (x86)\Windows Embedded 8 Standard\Developer Content\SDK\inc\ewfapi.h

C:\Program Files (x86)\Windows Embedded 8 Standard\Developer Content\SDK\x86\Write Filter\ewfapi.lib

C:\Program Files (x86)\Windows Embedded 8 Standard\Developer Content\SDK\amd64\Write Filter\ewfapi.lib

Prima di utilizzare queste interfacce è bene ricordare che:

L’interfaccia è offerta soltanto per una programmazione WIN32 in C/C++;

Le chiamate di questa interfaccia, a run-time, hanno effetto soltanto se utilizzate da un utente con privilegi di amministratore;

Molti dei comandi che riguardano il ***filtro-EWF*** hanno effetto soltanto dopo un restart del sistema;

Tutte le configurazioni e i comandi che si possono dare al ***filtro-EWF*** dall’applicativo a riga comando o da quello grafico, possono essere gestiti attraverso questa interfaccia programmabile.

Rimandiamo l’approfondimento delle definizioni, delle funzioni e della sintassi dei vari comandi ad un articolo dedicato in cui tratteremo, oltre a queste **API**, anche la possibilità di utilizzare comandi **Power-Shell** e **WMI** (Windows Management Instrumentation) locali o remoti.

## HORM (=Hibernate Once, Resume Many)

Una funzionalità interessante è quella data dalla possibilità di far lavorare il filtro di scrittura con il file di ibernazione del sistema. Quando si chiede al sistema di “ibernarsi” questi copia il contenuto attuale della RAM di lavoro in un file (***hiberfil.sys***) e poi spegne il dispositivo. Al riavvio, il sistema sa che sta tornando da un’ibernazione e quindi riprende il contenuto del file ***hiberfil.sys*** lo ricopia in RAM, richiede (se configurata) l’autenticazione dell’utente e quindi si ritrova nel medesimo punto dove era stata richiesta l’ibernazione. Questa funzionalità, legata alle qualità HW del dispositivo, permette un avvio molto rapido perché si saltano tutte le fasi di caricamento degli applicativi in memoria.

A questo punto con un comando da uno dei programmi di gestione preposti o con una funzione di programmazione si può chiede ad un filtro capace di gestire questa funzione: ***filtro-EWF*** e ***filtro-UWF*** di attivarla. Appena finito di salvare il contenuto della RAM su disco il filtro attiva la sua protezione sul disco dove ha salvato il file e quindi si ottiene che, da quel momento in poi, il sistema a tutti i riavvii si comporterà come al ritorno dalla stessa ibernazione.

Questa funzionalità si chiama **HORM** (=Hibernate Once, Resume Many) e, vista la difficoltà di trovare una traduzione soddisfacente, la chiameremo così anche in italiano.

L’**HORM**, quindi, permette un avvio veloce del dispositivo perché gli applicativi sia di sistema che non, sono già caricati in memoria. Bisogna stare attenti a tutte quelle operazioni che sono state fatte nella procedura di caricamento “normale” e che non vengono effettuate in questo caricamento diretto della memoria: inizializzazioni hardware, autenticazioni, montaggi di periferiche, ecc...

A differenza da Windows Embedded Standard 2009, in Windows Embedded 8 Standard la videata di ripristino del sistema quando si riavvia da un’ibernazione può essere soppressa o modificata.

Se le politiche di operatività del vostro dispositivo prevedono uno spegnimento ed avete necessità di un avvio immediato, l’**HORM**, con le raccomandazioni già citate, può essere una buona soluzione. Tenete comunque presente che per scrivere il file di ibernazione bisogna disporre di tanto spazio quanta è la RAM del sistema (anche se durante le sessioni ne viene utilizzata meno) questo punto di attenzione potrebbe sembrare eccessivo, ma molti dispositivi embedded utilizzano dischi-flash e RAM di ridotte dimensioni e quindi non sempre hanno 1 o 2 Gb di spazio per l’Hiberfil.sys. Una seconda osservazione è legata al tempo di caricamento di un file di 1 o 2Gb da una flash che potrebbe risultare comparabile con il tempo di caricamento del sistema e successivamente degli applicativi.

E’ da notare, inoltre, che questa tecnologia prevede che, se si hanno più di una partizione nel sistema, tutte le partizioni debbano essere protette da un ***filtro-EWF*** o ***filtro-UWF*** questa particolarità, comunque può essere gestita in modo da proteggere soltanto una partizione “montando” e “smontando” opportunamente le partizioni che NON si vogliono proteggere. Per maggiori informazioni, fare riferimento all’articolo:

<http://msdn.microsoft.com/en-us/library/dd143253(v=winembedded.5).aspx> *(in inglese)*

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

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