Microsoft | Desktop Virtualization

Windows Thin PC

Enhanced Write Filter

Contents

June 2011

Enhanced Write Filter

Enhanced Write Filter (EWF) lets you write-protect a run-time image. By redirecting all write requests to RAM, EWF enables the run-time image to maintain the appearance of a writable run-time image.

In this Section

EWF Overview

Provides an overview of Enhanced Write Filter. This overview describes the EWF architecture and the components that are required to support EWF

EWF Modes

Describes the different modes of EWF, and how to implement each type.

EWF Design Considerations

Describes the considerations you should make before configuring and deploying an EWF-protected run-time image.

EWF Overview

The Enhanced Write Filter (EWF) protects a volume from write access. EWF write-protects one or more partitions on a system.

EWF can be deployed on many media types and configurations.

Important

EWF cannot protect media that is marked as “Removable”. EWF can only protect media that is marked as “Fixed”.

The two major components for EWF are the EWF Overlay and the EWF Volume:

- **EWF Overlay:** EWF protects the contents of a volume by redirecting all write operations to another storage location called an overlay. An EWF overlay is located in RAM. An overlay is conceptually similar to a transparency overlay on an overhead projector. Any change that is made to the overlay affects the picture as it is seen in the aggregate. However, if the overlay is removed, the underlying picture remains unchanged.
- **EWF Volume:** There are two different EWF modes based on the configuration of the EWF volume.

EWF Mode	EWF Overlay Location	EWF Volume Location	Description
RAM	In RAM	Created on disk in unpartitioned space	EWF stores overlay information in RAM. When the system is restarted, all the information in the overlay is discarded. Use EWF RAM types on systems if you want to discard any write information after restart, or to delay writing the overlay to the media. For more information, see EWF RAM Mode.
RAM Reg	In RAM	In system registry	Similar to EWF RAM types, RAM Reg overlays store overlay information in RAM. However, the configuration information about EWF is not stored in a separate EWF volume, but within the registry. Use EWF RAM Reg types on media that does not support changing the partition structure of the media. For more information, see EWF RAM Reg Mode.

In this Section

EWF System Requirements

Before you deploy an EWF-protected run-time image, make sure that the system meets the following requirements.

Partitioning Requirements for EWF Setup

- For EWF RAM mode, verify that the disk on which you configure Enhanced Write Filter (EWF) has no more than three primary partitions. Disks are limited to a total of four primary partitions. If you are using logical partitions, free space for EWF must be contained within the extended partition.

  Verify that there is at least 8 MB of available space in one of the following areas:

  - Immediately after a primary partition
  - In an extended partition

    Note

    EWF creates the EWF volume in an extended partition, even if there is space at the end of the disk outside the extended partition.

This space is not required if you are using RAM REG mode.

For more information about EWF volumes, see EWF Volume Configuration.

- Keep your disk partition configurations simple. Complex disk partitioning can sometimes prevent EWF from setting up correctly. Whenever possible, use a single disk that has a single primary partition on it when you configure EWF for the first time.

EWF Media Requirements

- EWF does not protect dynamic disks. All protected volumes must be configured as basic disks.
- You cannot create an EWF volume on media that is marked as removable, such as CompactFlash. Some CompactFlash manufacturers provide utilities to mark the media as fixed. This lets you create an EWF volume. Optionally, you can use EWF RAM Reg mode to bypass the requirement for an EWF volume. For more information, see Configure EWF RAM Reg Mode.

EWF Architecture

Enhanced Write Filter (EWF) is a lower filter driver in the volume stack. It is located between file systems and the class drivers that interface with physical disks.

EWF Manager (EWFMGR) is a console application that provides a command-line interface for managing EWF.

The Enhanced Write Filter driver, Ewf.sys, redirects write I/O Request Packets (IRPs) to the EWF overlay. The EWF overlay is a write cache that can be stored in RAM. Read-only IRPs cause the EWF driver to search for a match in the current overlay stack. If the sector is found in the overlay, data from the overlay is returned. Otherwise, data from the protected volume is returned.

The EWF volume stores metadata about the current EWF configuration. For overlays, it also stores information about the protected volume.

For more information about EWF types and overlay configurations, see EWF Modes.

For more information about the EWF volume, see EWF Volume Configuration.

EWF Volume Configuration

When EWF is configured during the EWF installer process, it creates a small partition on the media. This partition is called the EWF volume. The EWF volume stores configuration information about the EWF-protected volumes on the device. This includes the number of protected volumes, protected volume sizes, and any of the overlay levels. Only one EWF volume is created on your device, regardless of how many disks are in the system.

The following list shows the EWF volume configuration requirements:

- For EWF RAM mode, the EWF volume is less than 64 KB. The EWF volume for RAM-based overlays stores additional boot commands that are applied when the system restarts, for example, EWF enable and disable.
- For EWF RAM Reg mode, the configuration information that is stored in the EWF volume is instead stored in the registry. RAM Reg modes are useful when you cannot create a new partition for the EWF volume.

During the EWF installer phase, EWFcfg.dll creates the EWF volume on your media. The EWF volume is created in one of the following two types of spaces on your media:

- Immediately after a primary partition
- Within the empty area in an extended partition

Note

If an extended partition is available on the system, EWF will always try to create the EWF volume within the empty area in the extended partition, even if there is additional free space at the end of the disk. If you have an extended partition, make sure that there is space within the partition for the EWF volume.

Note

Disk overlays must be disabled before re-deploying a pre-EWF installer runtime on the drive. If they are not disabled, EWF will be unable to delete and re-create these EWF Volumes during the EWF installer phase. If this step is missed then the error “missing or corrupted hal.dll” will be thrown during later starts of the system.

Make sure that your media can support creating an EWF volume. Some media, such as CompactFlash devices that are marked as removable, do not enable you to create new partitions. However, if the media is marked as fixed, you can create more than one partition on it. Some CompactFlash manufacturers supply utilities to mark the media as fixed.

Additionally, to prevent the operating system from changing drive letters that are assigned to existing volumes, use the following rule when you create the EWF volume: If an extended volume exists on the drive with sufficient space, the EWF volume is created as a logical volume. Otherwise, the EWF volume is created in a primary volume.

If you cannot create a new partition or change the partition structure of your media, consider implementing EWF RAM Reg mode. For more information, see EWF RAM Reg Mode.

The following list shows some common problems when configuring the EWF volume:

- There are already four primary partitions on the disk. Disks can have no more than four primary partitions.
- There is no space immediately after the primary partition.
- There is no space within the extended partition. The EWF volume cannot be created immediately after an extended partition.
- There is an existing EWF volume that was not deleted.

EWF Modes

There are two different modes of EWF based on the implementation of the EWF volume. EWF stores the overlay data in RAM. The EWF volume can exist in unpartitioned space on the media, or within the system registry. For more information, see EWF Overview.

Select the mode of the EWF overlay when you configure the Enhanced Write Filter with HORM settings. EWF RAM Reg mode requires additional configuration settings after you deploy the run-time image.

In this Section

EWF RAM Mode

Describes when to use and how to configure EWF RAM mode. EWF RAM mode stores overlay data in RAM, and stores the EWF volume in unpartitioned space on the disk.

EWF RAM Reg Mode

Describes EWF RAM Reg mode, how it differs from EWF RAM mode, and additional configuration details. EWF RAM Reg mode stores overlay data in RAM, and stores the EWF volume in the system's registry.

EWF RAM Mode

EWF RAM mode uses the following configuration:

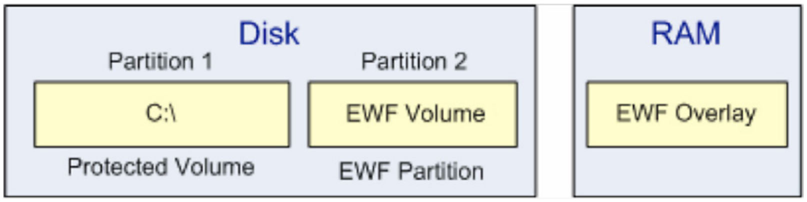
- The EWF Overlay type is stored in RAM.
- The EWF volume is stored on disk.

The EWF RAM overlay stores the write cache in RAM. When EWF RAM mode is configured during the EWF installer phase, the EWF volume is created in available space on media. The EWF volume stores the EWF master volume table and an overlay stack that points to the overlay data in system memory. When you shut down the target system, the overlay data in system memory is lost. EWF RAM mode supports only one overlay level.

Use EWF RAM mode in the following scenarios:

- Protecting data on a read/write volume from being altered or corrupted.
- Deploying a run-time image on a stateless device.
- Deploying a run-time image on a device without persistent read/write storage.

The following diagram shows an example EWF RAM mode configuration.



EWF RAM overlays require additional RAM to store the write cache. RAM is not pre-allocated by EWF. Therefore, EWF uses free RAM until the system runs out of memory. RAM requirements for the EWF RAM overlay vary. They depend on the amount of write operations that are made to the overlay.

For more information, see Configure EWF RAM Mode.

Configure EWF RAM Mode

EWF RAM mode stores overlay data in system memory. Because the overlay is stored in RAM, your device may require additional RAM, depending on how many write operations are made to the system. Also, because the data cache is stored in volatile memory, the data cache is lost when the system restarts.

▶ To configure EWF RAM mode

1. Deploy the image and start the device. The EWF installer adds the EWF Components to the device. The Ewfcfg.exe tool is a proxy to Ewfcfg.dll, which handles EWF setup. Ewfcfg.dll reads and applies the configuration settings from the answer file.
2. After the install process is complete, verify the status of EWF by typing the following command line.

   ```
   ewfmgr -all
   ```

   By default, EWF should report that it is disabled on all drives.
3. Enable EWF by typing the following command line:

   ```
   ewfmgr c: -enable
   ```

4. Restart the system.
5. Verify that EWF is correctly configured by typing the following command at a command prompt:

   ```
   ewfmgr c:
   ```

   Ewfmgr.exe should report that EWF is enabled and provide a current status of the EWF RAM overlay.

EWF RAM Reg Mode

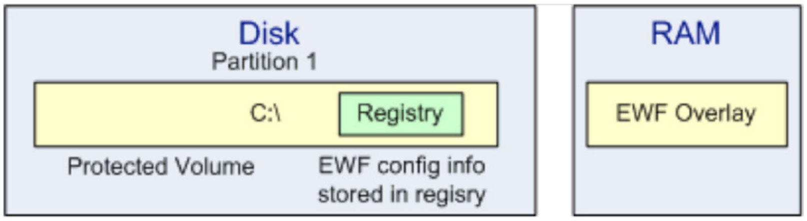
EWF RAM Reg mode resembles EWF RAM mode. It differs from the standard EWF RAM mode only in that the configuration information that is stored in the EWF volume is instead stored in the system's registry. EWF RAM Reg mode uses the following configuration:

- The EWF Overlay is stored in RAM.
- The EWF volume is stored in the system's registry.

Use RAM Reg mode in the following scenarios:

- Protecting media that has only a single partition or media that cannot be partitioned.
- Protecting removable media, such as CompactFlash, USB, or IEEE 1394 devices.
- Protecting media that does not support standard RAM overlays.
- Reducing the number of write operations that are made to write-sensitive devices, such as CompactFlash.

The following illustration shows an example of how EWF RAM Reg mode is configured on your device.



Because the EWF volume information is kept in the registry, if the registry is write-protected, any attempts to disable EWF will fail. There are three ways that you can disable EWF in EWF Ram Reg mode:

- Disable EWF, and then commit the overlay to the protected volume.

  For example:

  ```
  ewfmgr c: -disable
  ewfmgr c: -commit
  ```

⚠ Warning

You must reboot the system in the previous scenario because the overlay is written to the protected volume on the next system restart.

-Or-

- Use the CommitandDisable command to disable EWF and commit the overlay.

  For example:

  ```
  ewfmgr c: -commitanddisable
  ```

⚠ Warning

You must reboot the system in the previous scenario because the overlay is written to the protected volume on the next system restart.

-Or-

- Use the -live option of the CommitandDisable command to disable EWF and commit the overlay to the protected volume without having to restart the system.

  For example:

  ```
  ewfmgr c: -commitanddisable -live
  ```

For more information, see EWF Manager Commands.

For more information about how to configure EWF RAM Reg mode, see Configure EWF RAM Reg Mode.

You can also protect multiple volumes by using EWF RAM Reg modes.

Configure EWF RAM Reg Mode

EWF RAM Reg mode stores overlay data in RAM and EWF configuration information in the registry. Because EWF RAM overlays store the data cache in RAM, your device may require additional RAM. Also, because the data cache is stored in volatile memory, the data cache is lost when the system restarts.

Additionally, because EWF configuration information is stored in the registry, it is typically protected by EWF. You must use the **ewfmgr -commitanddisable** command to apply any changes to EWF.

▶ To configure EWF RAM mode

1. Deploy the image and start the device. The EWF installer adds the EWF Components to the device. The Ewfcfg.exe tool is a proxy to Ewfcfg.dll, which handles EWF setup. Ewfcfg.dll reads and applies the configuration settings from the answer file.
2. After the install process is complete, verify the status of EWF by typing the following command line.

   ```
   ewfmgr -all
   ```

   By default, EWF should report that it is disabled on all drives.
3. Enable EWF by typing the following command line:

   ```
   ewfmgr c: -enable
   ```

4. Restart the system.

5. Verify that EWF is correctly configured by typing the following command at a command prompt:

   ```
   ewfmgr c:
   ```

   Ewfmgr should report that EWF is enabled and provide a current status of the EWF RAM overlay.

   Note

   If you are using an intermediary hard disk, copy the run-time image to the CompactFlash device after verifying that EWF is configured correctly.

EWF Design Considerations

This section examines common design considerations for Enhanced Write Filter (EWF).

In this Section

EWF Maintenance and Management

Describes how to manage EWF by using the EWFMGR command. This lets you install updates to a run-time image that is protected by EWF.

EWF Performance Considerations

Describes ways that you can improve the performance of EWF.

EWF and File-Based Write Filtering

Describes how the EWF and File-Based Write Filtering (FBWF) are related.

EWF Maintenance and Management

You can use the EWF Manager tool to enable or disable EWF and to commit overlays to the protected volume. You can use the EWF Manager command to disable EWF so that you can apply updates to your run-time image.

In this Section

EWF Manager Commands

Describes the command syntax for the EWF Manager command.

Install Updates on an EWF-Protected Run-Time Image

Describes the process to follow and the considerations to make when you install applications or QFEs to a protected run-time image.

Write Filters and Automatic Adjustment of Daylight Saving Time

Describes the recommended approach for handling Daylight Saving Time when using write filters.

EWF Manager Commands

The EWF Manager console application is used to control Enhanced Write Filter (EWF). EWF Manager uses the following syntax.

Syntax

EWFMGR [`<volume-name>*`](optional) [`options`]

Parameters

*<volume-name>**

Specifies the volume path. This is an optional parameter that is used for protected volume configuration mode.

The volume name can be either a drive letter (for example, C:), a device name (for example, \Device\Harddiskvolume), or a volume GUID path in the form of "\\?\Volume{GUID}\" where GUID is a globally unique identifier (GUID) that identifies the volume (for example, \\?\Volume{26a21bda-a627-11d7-9931-806e6f6e6963}\).

Referencing volumes using GUIDs is more reliable because you avoid factors that make it difficult to identify a volume. For example, two volumes having the same label, a volume having no drive letter or label, and drive letters changing as volumes are added to and removed from the computer. For more information, see Naming a Volume. To retrieve the GUID volume path for a volume, see the GetVolumeNameForVolumeMountPoint Function.

Notice that the name is not the volume label that Windows Explorer displays before the drive letter.

The default behavior is to display information about the EWF volume configuration if no <*volume-name*> is specified. To view the status of the protected volume, specify the drive letter for the protected volume, for example, `ewfmgr c:`

```
ewfmgr \\?\GLOBALROOT\Device\HarddiskVolume1
```

Note

Multiple volumes may be specified. The *volume-name* may be identified using a device name, a GUID volume path, or a drive letter and colon.

Note

You can use the **-all** command in place of the *volume-name* parameter to perform the specified action on all volumes.
For example, to enable EWF for all volumes, type: `ewfmgr -all -enable`

Important

Disk-backed overlay operations are not supported.

options

Specifies the EWF volume boot options.

The following commands are used to manage protected volume configuration: **Disable**, **Enable**, **Commit**, **CommitandDisable**, **Persist**, **ActivateHORM**, **DeactivateHORM**, and **Nocmd**.

Remarks

The following table shows the Enhanced Write Filter (EWF) Console Manager Application tool boot commands.

Boot command	Description
All	Performs a specified command on all protected volumes.
Commit	Commits all current level data in the overlay to the protected volume, and resets the current overlay level to 1. The **Commit** command can be combined with the **Disable** command to commit and then disable. The overlay is written to the protected volume on the next system restart. Committing the overlay can affect the speed of the boot process.
CommitandDisable	Commits all current level data in the overlay to the protected volume and disables the overlay. The overlay is written to the protected volume on the next system restart. Committing the overlay can affect the speed of the boot process. You can use the **-live** command for both EWF RAM and EWF RAM Reg modes to immediately commit the overlay to the protected volume and disable the overlay without having to restart the system. For example, `ewfmgr c: -commitanddisable –live`

Boot command	Description
Disable	Disables the overlay on the specified protected volume. **Important** When you use the Disable command in RAM REG mode, changes are not persisted to the registry. You must use the -**CommitAndDisable** command when in RAM REG mode.
Enable	Enables the write filter so that data that is written to the protected media is cached in the overlays. The current overlay level becomes 1 as soon as EWF is started, and a new overlay is created at level 1.
NoCmd	Clears the current pending command.
Persist	Specifies a 64-byte field that persists throughout all overlays for a specific protected volume. The Persist command enables you to store EWF-specific or application state information and can be useful when you update a device. Persistent data is stored in the EWF volume store.
ActivateHorm	Enables HORM. **Important** HORM has a requirement that all volumes must either be protected with EWF or be in unmounted state when the Hibernate Once occurs. This is to prevent state synchronization problems. Each Resume from hibernation expects the entire system to be in exactly the same state as when the Hibernate Once occurred.
DeactivateHorm	Disables HORM.

Because EWF Manager commands are executed on the next start, you must restart the system for a command to take effect.

Example

Microsoft.Win32.RegistryKey#4

The following examples refer to a system on which EWF RAM REG mode is configured to protect drive C.

The following example shows you how to check the EWF status and format.

```
ewfmgr c:
```

EWF Manager displays the following result.

```
Protected Volume Configuration
   Type            RAM(REG)
   State           DISABLED
   Boot Command    NO_CMD
     Param1         0
     Param2         0
  Volume ID       58 55 BF A4 00 00 50 06 00 00 00 00 00 00 00 00
  Volume Name     “\\?\GLOBALROOT\Device\HarddiskVolume2” [C:]
  Max Levels      1
  Clump Size      512
  Current Level   N/A

  Memory used for data 0 bytes
  Memory used for mapping 0 bytes
```

The following example shows you how to enable EWF for drive C.

```
ewfmgr c: -enable
```

EWF Manager displays the **Enable** command as pending. The command does not execute until the next restart. EWF Manager displays the following result.

```
*** Enabling overlay
Protected Volume Configuration
  Type            RAM (REG)
  State           DISABLED
  Boot Command    ENABLE
    Param1        0
    Param2        0
  Volume ID       58 55 BF A4 00 00 50 06 00 00 00 00 00 00 00 00
  Volume Name     “\\?\GLOBALROOT\Device\HarddiskVolume2” [C:]
  Max Levels      1
  Clump Size      512
  Current Level   N/A

  Memory used for data 0 bytes
  Memory used for mapping 0 bytes
```

The following example shows you how to check the status type of the EWF volume.

```
ewfmgr -all
```

EWF Manager displays the following result.

```
  Type            RAM (REG)
  State           ENABLE
  Boot Command    NO_CMD
    Param1        0
    Param2        0
  Volume ID       58 55 BF A4 00 00 50 06 00 00 00 00 00 00 00 00
  Volume Name     “\\?\GLOBALROOT\Device\HarddiskVolume2” [C:]
  Max Levels      1
  Clump Size      512
  Current Level   1

  Memory used for data 9910272 bytes
  Memory used for mapping 12288 bytes
```

Note

If EWF is disabled, the **current level** is shown as **N/A**.

Install Updates on an EWF-Protected Run-Time Image

Because Enhanced Write Filter (EWF) prohibits write access to a volume, there are additional considerations to make when you want to install an application or update your run-time image.

To disable the current EWF overlay, use the **ewfmgr disable** command. After you disable the EWF overlay, you must restart the system and then install your application.

Note

> For EWF RAM Reg mode, you must use the **ewfmgr commitanddisable** command to disable EWF. The **ewfmgr commitanddisable** command applies all changes in the overlay to the protected volume, even those that are unrelated to your application installation or update. To minimize additional write operations to your volume, you can clear the overlay cache by restarting the device before you install any applications. For more information, see EWF RAM Reg Mode.

To apply an update, you must use one of the following procedures.

▷ **To update a run-time image that is protected by EWF RAM mode:**

1. Use EWF Manager to disable the overlay by typing the following command:

   ewfmgr c: -disable
2. Restart the system.
3. Install the application or update.
4. Wait for the install to complete and restart the computer if required.
5. Re-enable the EWF overlay by using the following command:

   ewfmgr c: -enable
6. Restart the system to re-enable the EWF overlay.

▷ **To update a run-time image that is protected by EWF RAM Reg mode:**

1. Restart the device to clear the RAM overlay.
2. Commit the overlay to the protected volume and disable the EWF overlay by typing the following command:

   ewfmgr c: -commitanddisable

   Because RAM Reg modes store EWF configuration data in the registry, you must commit the disable change to the protected run-time image. For more information, see Configure EWF RAM Reg Mode.
3. Restart the system to disable the overlay.
4. Install the application or update.
5. Wait for the install to complete and restart the computer if required.
6. Enable the EWF overlay by typing the following command:

   ewfmgr c: -enable
7. Restart the system to re-enable the EWF overlay.

Write Filters and Automatic Adjustment of Daylight Saving Time

Automatic Adjustment of daylight saving time (DST) is incompatible with the File-Based Write Filter (FBWF) and the Enhanced Write Filter with HORM (EWF). If Automatic Daylight Saving is enabled while FBWF or EWF is enabled, the system clock will either fall back or spring forward every time the computer is restarted.

The recommended approach to solving issues with DST when using write filters is to change the CMOS clock to use UTC.

▷ **Change the CMOS clock to use UTC:**

1. Make sure that EWF and FBWF are disabled.
2. Configure the Windows systems Time Zone to the desired value (for example, Pacific Time Zone).
3. Add the RealTimeIsUniversal key DWORD value to the registry at HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\TimeZoneInformation\.
4. Change the value of the RealTimeIsUniversal key to 1.
5. Restart the system and go into the BIOS.
6. Set the CMOS clock to the current UTC time and date. For example, if you are in Pacific Standard Time Zone, and it is 11:00 AM, then you would set the CMOS clock to 7:00 PM as there is 8 hours difference.
7. Restart the system and enable EWF or FBWF.

EWF Performance Considerations

This section includes topics about how you can improve the performance of your run-time image with Enhanced Write Filter (EWF) enabled.

In this Section

Change the Location of the Event Log

Describes how to change the location of the system event log to a nonprotected volume.

Change the Location of the Pagefile

Describes how to change the location of a system pagefile to a nonprotected volume.

Change the Location of the Temporary Files Folder

Describes how to change the location of the temporary files directory to a nonprotected volume.

Disable Last Access Time Stamps

Describes how to disable NTFS last access time stamps.

Disable Prefetch

Describes how to disable **Prefetch** to improve EWF performance.

Disable SuperFetch

Describes how to disable **SuperFetch** to improve EWF performance.

Change the Location of the Event Log

To improve the performance of Enhanced Write Filter (EWF) on a system that uses an event log, you can relocate the event log to an alternative partition that is not EWF-protected. This requires at least two partitions: one partition that EWF protects, and another partition that is writable.

▷ **To change the location of the event log:**

1. To change the location of an event log to an unprotected volume, you must update the registry of the run-time image. Modify the following three registry keys, and change the event log to an unprotected volume.

   Key Name: **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\EventLog\Application\**

   Value Name: **File**

   Type: **REG_EXPAND_SZ**

   Value: <*Volume Name and Path*>**\AppEvent.evt**

   Key Name: **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\EventLog\Security\**

   Name: **File**

   Type: **REG_EXPAND_SZ**

   Value: <*Volume Name and Path*>**\SecEvent.evt**

   Key Name:
   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\EventLog\System\

   Name: **File**

   Type: **REG_EXPAND_SZ**

   Value: <*Volume Name and Path*>**\SysEvent.evt**

2. In the **Value** field, change the path of the event file to a nonprotected volume.

Change the Location of the Pagefile

To improve the performance of Enhanced Write Filter (EWF) on a system that uses a pagefile, you can relocate the system pagefile to an alternative partition that is not EWF-protected. This requires at least two partitions: one partition that EWF protects, and another partition that is writable.

▷ **To change the location of the pagefile:**

1. After a run-time image is deployed, edit the following registry key:

   Key Name: **HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management**

   Value Name: **PagingFiles**

   Type: **REG_MULT_SZ**

   Data: **C:\Pagefile.sys 150 500**

2. In the **Data** field, change the path and file name of the pagefile, together with the minimum and maximum file size values (in megabytes).

Change the Location of the Temporary Files Folder

You can improve Enhanced Write Filter (EWF) performance on a run-time image by changing the temporary files location to a different partition that is not EWF-protected. By default, temporary Internet files are stored in the %USERPROFILE%\AppData\Local\Microsoft\Windows folder.

▷ **To change the location of the temporary files folder:**

1. Modify one or both of the following registry keys on your run-time image:
   - Key: **HKEY_CURRENT_USER\Software\Microsoft\Windows\Current Version\Explorer\User Shell Folders**
     Name: **Cache**
     Type: **REG_EXPAND_SZ**
     Value: *<path_to_a_folder_on_an_unprotected_volume>*
   - Key: **HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders**
     Name: **Cache**
     Type: **REG_EXPAND_SZ**
     Value: *<path_to_a_folder_on_an_unprotected_volume>*
2. You can also redirect the TMP and TEMP folders to an unprotected volume by modifying the following registry settings:
   - Key: **HKEY_CURRENT_USER\Environment**
     Name: **TEMP**
     Type: **REG_SZ**
     Value: *<path_to_a_folder_on_an_unprotected_volume>*
   - Key: **HKEY_CURRENT_USER\Environment**
     Name: **TMP**
     Type: **REG_SZ**
     Value: *<path_to_a_folder_on_an_unprotected_volume>*

Disable Last Access Time Stamps

If you are using an NTFS file system, you can increase the performance of EWF by disabling the last access date/time stamps.

▷ **To disable Last Access timestamps:**

- Create the following registry key on your run-time image:
  Key Name: **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\FileSystem**
  Name: **NtfsDisableLastAccessUpdate**
  Type: **REG_DWORD**
  Value**: 1**

Note

If you need to re-enable Last Access Time Stamps, change the value of the NtfsDisableLastAccessUpdate key to 0, or delete the NtfsDisableLastAccessUpdate key. For more information, see NtfsDisableLastAccessUpdate.

Disable Prefetch

Enhanced Write Filter (EWF) performance can sometimes be improved by disabling Prefetch. Prefetch is a tool that is intended to improve operating system and application startup performance. It does this by loading application data into memory before it is demanded.

When you are using EWF with a RAM overlay to protect the boot volume, Prefetch is unable to persist its data from startup to start-up. Under these conditions, Prefetch tries to compute and save new data files every time that the system starts using EWF resources with, potentially, no benefit.

▶ To disable Prefetch

- Update the **EnablePrefetcher** registry key in your run-time image:

  Key: **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager \Memory Management\PrefetchParameters**

  Name: **EnablePrefetcher**

  Type: **REG_DWORD**

  Value: **0**

The **EnablePrefetcher** key has the following values:

0 = Disabled

1 = Application start prefetching enabled

2 = Boot prefetching enabled

3 = Application start and boot enabled

To disable Prefetch, set the value to 0.

Disable SuperFetch

Enhanced Write Filter (EWF) performance can sometimes be improved by disabling SuperFetch. SuperFetch is a tool that is intended to improve operating system and application startup performance. It does this by loading application data into memory before it is demanded. SuperFetch improves on Prefetch by monitoring which applications you use the most and preloading those into your system memory so they will be ready when you need them.

When you are using EWF with a RAM overlay to protect the boot volume, SuperFetch is unable to persist its data from startup to startup. Under these conditions, SuperFetch tries to compute and save new data files every time that the system starts using EWF resources with, potentially, no benefit.

▶ To disable SuperFetch

- **Update the EnableSuperfetch** registry key in your run-time image:

  Key: **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager \Memory Management\PrefetchParameters**

  Name: **EnableSuperfetch**

  Type: **REG_DWORD**

  Value: **0**

The **EnableSuperfetch** key has the following values:

0 = Disabled

1 = Application start SuperFetching enabled

2 = Boot SuperFetching enabled

3 = Application start and boot SuperFetching enabled

To disable SuperFetch, set the value to 0.

EWF and File-Based Write Filtering

Thin PC provides two write filters: File-Based Write Filtering (FBWF), which operates at the file level, and Enhanced Write Filter (EWF), which operates at the sector level. In most cases, FBWF is the better choice. However, only EWF fully supports NTFS.

The following NTFS features are supported in EWF, but not in FBWF:

- File locking and unlocking
- File ID in NTFS
- Reparse points
- Quota
- Hard links
- Opportunistic lock
- File compression and encryption

FBWF does not provide the following EWF function:

- Live commit and disable

To use EWF and FWBF together in the same OS image, enable EWF on one volume and FBWF on another volume.