Microsoft | Desktop Virtualization

Windows Thin PC

File-Based Write Filter

Contents

June 2011

File-Based Write Filter (FBWF)

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

In this Section

FBWF Overview

Provides an overview of File-Based Write Filter. This overview describes FBWF operation and architecture.

FBWF Features

Describes the new features that were introduced in FBWF.

FBWF Installation and Configuration

Describes the tools and procedures for installing and configuring FBWF.

FBWF Manager

Provides a guide to using FBWF Manager for quickly prototyping and integrating FBWF at design time.

FBWF Overview

Writing to storage media may be unwanted or impossible in Thin PC devices. FBWF redirects all writes targeted for protected volumes to a RAM cache called an overlay.

Because the File-Based Write Filter cannot be enabled while the system is running, the FBWF differentiates between the current session (since the last system restart) and the next session (after the next system restart).

In this Section

Advantages of FBWF

Outlines the advantages of FBWF.

FBWF Operation

Provides a high-level description of how FBWF works.

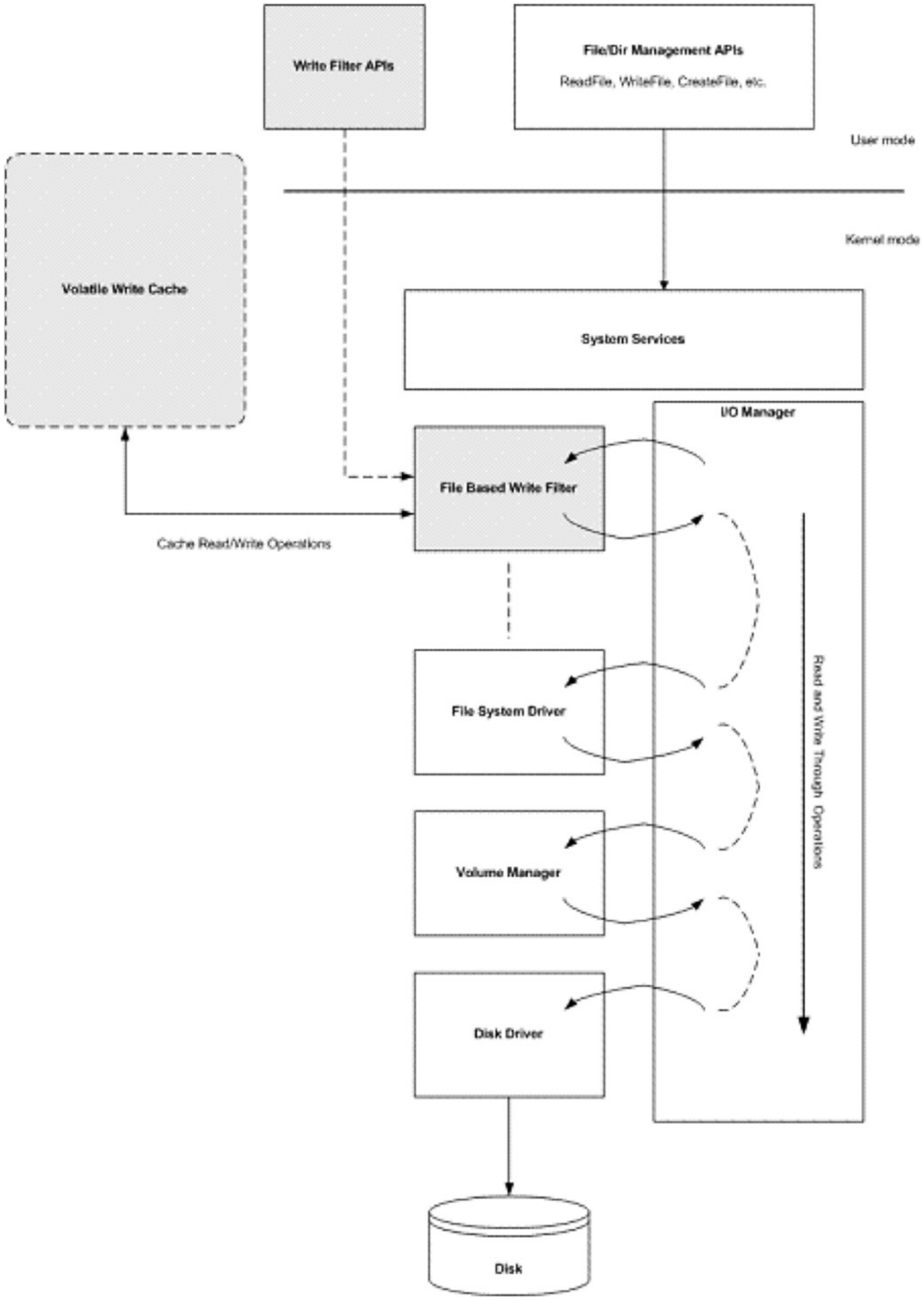
Advantages of FBWF

FBWF prevents write requests to one or more volumes with enhanced protection and caches all write requests to volumes with enhanced protection in an overlay cache on a volatile store. FBWF operates at the file level instead of the sector level, and provides the following features:

- File and Directory Management Transparency
- Selective Write-Through
- Selective Commits and Restores
- Improved Overlay Memory Use

FBWF Operation

FBWF operates at the file level to redirect all write requests targeted for protected volumes to overlay cache. The following illustration shows the relationship of FBWF to file system I/O and subsystem components.



FBWF Features

FBWF includes write filter transparency to applications; selective write-through, commit, and restore; improved overlay memory use; and enhanced APIs.

In this Section

File and Directory Management Transparency

Describes file and directory management functions available to applications.

Selective Write Through

Describes how to enable writes to specified underlying protected volumes.

Selective Commits and Restores

Describes how to commit writes from overlay cache to underlying protected volumes, and how to discard changes in overlay cache and restore the view to the underlying protected volumes.

FBWF Memory Optimization

Describes how FBWF optimizes the use of overlay cache.

File and Directory Management Transparency

Applications and OS components make file and directory management calls without knowledge that writes are being cached by FBWF. FBWF provides a composite view of the file system without imposing limitations on user mode applications. Applications perform explicit file I/O through functions such as **CreateDirectory**, **CreateFile**, **ReadFile**, and **WriteFile**. These calls appear to interact with the file system but may result in overlay cache I/O.

The following directory management operations may result in error:

- Moving files across protection boundaries, for example, from a protected volume to an unprotected volume.
- Committing deleted files or new directories.

FBWF supports NTFS, FAT32, and exFAT file systems.

FBWF can protect formatted volumes only. FBWF cannot recognize unformatted volumes.

Files in FBWF overlay cache are not encrypted.

Selective Write Through

Selective Write Through enables writes to a predefined set of files and directories to reach the underlying protected volume.
Files and directories are designated for Selective Write Through by using the following:

- Image Configuration Editor during the OS build process.
- FBWF APIs at run time.
- FBWF Manager at run time in the development environment.

Selective Commits and Restores

Selective Commit moves a file (or file changes) from the overlay to the protected volume. Selective Commit is immediate and persists through restarts. Selective Restore discards an overlay file and restores the view of the underlying volume.
Both functions are useful for devices that are rarely restarted or devices that are shared across users.

The commit functionality has the following limitations:

1. You can only commit one file at a time to an existing folder.
2. You cannot commit deleted files.
3. You cannot commit folders.

FBWF Memory Optimization

FBWF conserves and reclaims memory in overlay cache. For example, FBWF frees overlay memory when files are deleted or reduced in size.

You can configure FBWF to use one of the following cache types:

- Pre-allocated – In this mode, FBWF sets aside memory during initialization equal to the specified overlay size. FBWF frees memory within its pre-allocated cache, so that only FBWF can use the reclaimed memory.
- Dynamic – In this mode, FBWF allocates memory when it is needed up to a limit equal to the overlay size. FBWF frees memory that can be used by all processes and services on the system.
- Dynamic-Compressed – Same as Dynamic but the overlay will be compressed to save space.

FBWF Installation and Configuration

When deploying FBWF, you must configure the following features that are normally disabled by the FBWF installer:

1. Disable disk defragmentation by adding the following registry key to your run-time image:

   Disable Background disk defragmentation:

   Key Name: **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\defragsvc**

   Name: **Start**

   Type: **REG_DWORD**

   Value: **4**
2. (Optional) Change the BootStatusPolicy setting to IgnoreAllFailures using BCDEdit. This setting is not critical to FBWF operation. You can apply the setting using the following command:

   ```
   bcdedit.exe /set {bootloadersettings} BOOTSTATUSPOLICY IgnoreAllFailures
   ```
3. (Optional) Set the FBWF cache threshold size (128mb for x86, 256mb for amd64). If you do not supply a value, the driver will assume defaults. You may apply this setting using the FBWF APIs or fbwfmgr.exe (For example, `fbwfmgr /setthreshold 128`). The description of the setting as it appears in the manifest is:

If not specified, the value will default to 128 MB for 32-bit. Valid MB ranges are 16-1024 for 32-bit systems.

In this Section

Configure FBWF at Runtime

Describes how to configure FBWF at run time.

FBWF Performance Testing

Describes how to monitor the FBWF overlay cache.

Configure FBWF at Runtime

Configure FBWF at run time by using FBWF Manager.

FBWF Performance Testing

Monitor the FBWF overlay cache usage under real-world scenarios to verify that the cache size is appropriate for the intended use of the system. Strike the optimal balance between the size allocated to the overlay, the RAM on the system, and the requirements of the applications. The FBWF provides the following ways to monitor FBWF overlay cache use:

- The command-prompt tool **fbwfmgr** by using the **/overlaydetail** parameter.
- The **FbwfGetMemoryUsageEx** API.

If the overlay cache is running out of space too quickly, test your configuration either with a larger cache or with preallocation disabled.

FBWF Manager

The FBWF Manager is a command-line tool that is included in Thin PC to quickly integrate and prototype the FBWF (File-Based Write Filter). FBWF Manager is intended primarily for design-time use, but can also be used to configure FBWF at run time.

FBWF Manager Command Line Syntax

The FBWF Manager command-line syntax follows:

```
fbwfmgr  [ /? | /help /[switch] | /displayconfig |
 /overlaydetail | /enable | /disable | /addvolume [volumename] |
 /removevolume [volumename] [1|0] |
 /addexclusion [volumename] [file or directory path] |
 /removeexclusion [volumename] [file or directory path] |
 /setthreshold [threshold] | /setcompression [1|0] |
 /setpreallocation [1|0] | /commit [volumename] [file path] |
 /restore [volumename] [file path] | /setsizedisplay [1|0] |
 /getvirtualsize [volumename] | /getactualsize [volumename]
```

The following table describes the command-line switches.

Switch	Description
?	Displays usage and help.
Addexclusion	Adds a write-through path of the exclusion list for next boot.
Volumename	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.

Switch	Description
Addvolume	Adds a volume to the protected volume list for the next boot.
Commit	Commits the changes that you made to the file to the underlying media. 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 volume must currently be protected. Otherwise, you receive the following error message: "The system cannot find the drive specified."
disable	Disables the write filter on the next restart.
displayconfig	Displays all configuration information for the write filter including a protected volumes list, the overlay configuration, and write-through paths. The command returns: • State – Indicates current filter state (enabled or disabled) and state for the next boot. • Protected Volumes – List of protected volumes including the current and the next boot state. • Compression – Current and next boot state for cache compression. • Threshold – Current and next boot values for the overlay cache threshold. • Write-Through Paths – Displays a complete list of active and next boot write-through paths. • Preallocation Status – Displays current and next boot status for cache preallocation.
enable	Enables the write filter on the next restart.
getactualsize	Displays the actual volume disk size information.
getvirtualsize	Displays the virtual volume disk size information.
help / [switch]	Displays help information for a specific FBWF Manager switch.
overlaydetail	Displays details on the current overlay contents for all protected volumes. The command returns: • Contents – Files and folders currently in the overlay for all protected volumes including sizes (size of data in overlay) and open file handles. • Memory Usage – Total amount of memory being consumed by the overlay.

Switch	Description
removeexclusion	Removes a write-through path from the exclusion list for next boot. 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.
removevolume	Removes a volume from the protected volume list for next boot and either preserves the exclusion list (0) or removes it (1).
restore	Discards the changes that you made to the file, that is, restores the files to its original contents from the underlying media. 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. It is acceptable that the file was deleted. In this case, it is recovered. Notice that the volume must currently be protected. Otherwise, you receive the following error message "The system cannot find the drive specified."
setcompression	Sets overlay compression as enabled (1) or disabled (0) for the next boot.
setpreallocation	Sets cache pre-allocation as enabled (1) or disabled (0) for the next boot.
setsizedisplay	Sets the size display mode to either virtual (1) or actual (0). The new mode takes effect after the next restart.
setthreshold	Sets the overlay threshold value for the next boot. The input field threshold is the overlay threshold in MB.

If no switch is provided, the FBWF Manager displays all the configuration information, exactly as the **displayConfig** switch.

FBWF APIs are provided for comprehensive management and configuration of devices in the field.

FBWF does not resolve substituted paths. Use complete file names instead.

FBWF Integration

FBWF interacts with other components. Precautions must be taken at run time to guarantee correct operation of interacting components.

In this Section

FBWF and Network Credentials

Describes how the Registry Filter persists registry keys necessary for Domain Participation and Terminal Services Client Access Licensing (TSCAL).

FBWF and USB Mass Storage Devices

Examines the effect of removing protected USB devices that contain open files.

Service FBWF-Protected Devices

Provides the procedure to update the images on FBWF-protected devices.

Write Filters and Automatic Adjustment of Daylight Saving Time

Provides a procedure for handling automatic adjustment of Daylight Saving Time when using FBWF or EWF.

FBWF Interoperability with Windows 7 Features

Examines how FBWF interacts with Windows 7 features.

FBWF and Network Credentials

If the target device environment includes Domain Participation or Terminal Services Client Access Licensing (TSCAL), add and configure Registry Filter.

Enable the Domain Participation and TSCAL services while FBWF is in a disabled state to establish the necessary registry settings. When the services are confirmed to be operational, enable FBWF. Registry Filter persists the necessary registry keys from session to session.

FBWF and USB Mass Storage Devices

If FBWF is applied to a USB mass storage device, removing the USB device while it contains open files results in unpredictable behavior.

Service FBWF-Protected Devices

Servicing the image on an FBWF-protected device requires special care, depending on the files being updated.

Caution

> When a write filter (EWF or FBWF) is enabled, Offline servicing must not be used. Performing such servicing may result in unpredictable behavior that varies from problems booting up the system to updates not being applied.

▷ **To modify files that are being used, to modify the registry, or in update scenarios that require restart**

1. Disable FBWF.
2. Restart.
3. Service the image. If installing an update, wait for the install to complete and restart the computer (if required).
4. Enable FBWF.
5. Restart.

To enable and disable FBWF, use either the FBWF APIs **FbwfEnableFilter** and **FbwfDisableFilter** or the following FBWF Manager commands:

```
fbwfmgr /enable
```

```
fbwfmgr /disable
```

For servicing scenarios where affected files are not in use, and where registry modifications are not required, use FBWF to commit the changes after they are applied. For example, use FBWF to commit updates to the signatures used for virus detection.

Commit file changes by using either the FBWF API **FbwfCommitFile** or the following FBWF Manager command:

```
fbwfmgr /commit [volumename] [path]
```

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.

FBWF Interoperability with Windows 7 Features

FBWF changes the following Windows 7 settings:

- Changes the boot loader settings to ignore improper shutdown checks.
- Turns off search indexing.
- Disables Prefetch.

Caution

Write filters and Windows Media Center do not operate well together as Media Center performs a lot of disk IO and fills up the EWF or FBWF cache.