| | | |
|---|---|---|
| Title | : | Use of Write Filter in BVM XP Embedded Systems |
| Date | : | 04-02-10 |
| Product | : | XP Embedded |
| Pages | : | 5 |
| Issue | : | 1 |
| Originator | : | Ian Middleton |

# Use of Write Filter in BVM XP Embedded Systems

## Contents



## 1 Overview

BVM supplies Embedded PC systems using XP Embedded supplied on a CompactFLASH card.

All BVM XP Embedded systems are supplied with a write filter software component in order to overcome limitations of CompactFLASH.

The XP Embedded write filter “catches” all data writes to the CompactFLASH thus:-

- Overcoming limited the write endurance of CompactFLASH.
- Overcoming slow write speeds compared to read speed.
- Protecting the file system and data during unexpected system shutdown.

The CompactFLASH writes are stored in system RAM, by the write filter, and when data is reread the contents of RAM is read rather than CompactFLASH. Application software running on the system will be unaware of the operation of the write filter.

This has the following consequences to system operation:-

- Excessive data writing will eventually consume all system RAM.
- If the system is shut down either intentionally or unexpectedly all system changes since last boot will be lost.

BVM supplies systems with the write filter **DISABLED** as most supplied systems will require further configuration before end user use.

The write filter must be **ENABLED** before final end user use or else the system can be prone to disk corruption due to improper shut down.

The write filter is managed via the command line EWFMGR utility.

Typing "**ewfmgr c:**" (from a CMD.EXE prompt) will display the write filter status

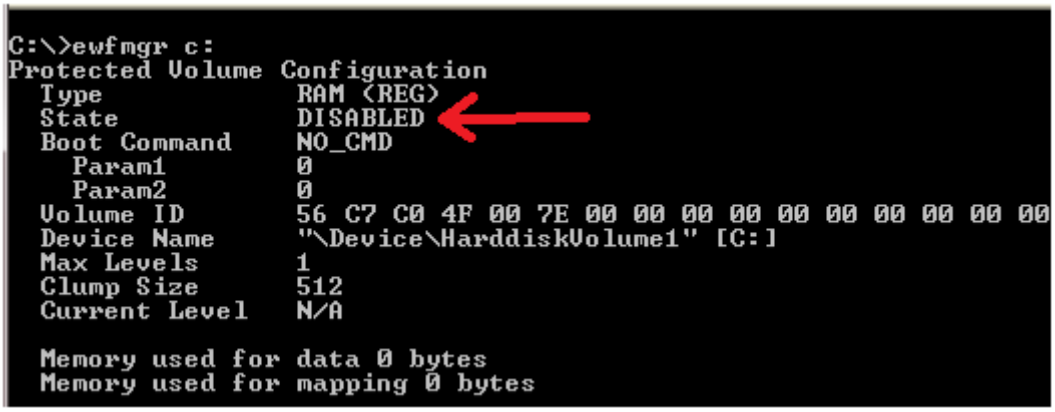


**Figure 1 Write Filter Disabled**

There are 5 basic operations of the write filter of interest to system developers. These are:

- Enabling the write filter.
- Discard changes and disable after next reboot.
- Commit changes and disable after next reboot.
- Commit changes and disable immediately.
- Commit changes next reboot.

### 1.1 Enabling the Write Filter.

Typing "**ewfmgr c: -enable**" will enable the write filter AFTER NEXT REBOOT.

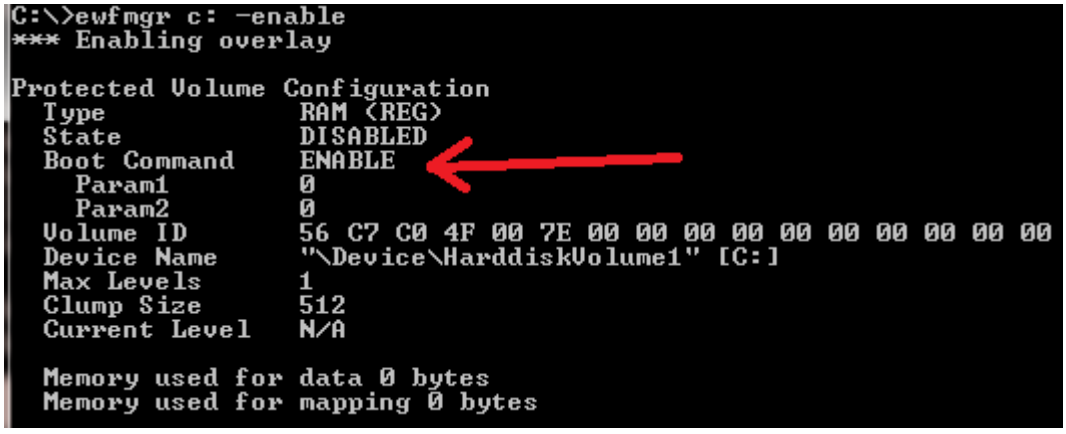


**Figure 2 Write Filter to Enabled Next Boot**

After shutdown and reboot the write filter will now be enabled. Typing "**ewfmgr c:**" will display the write filter status.

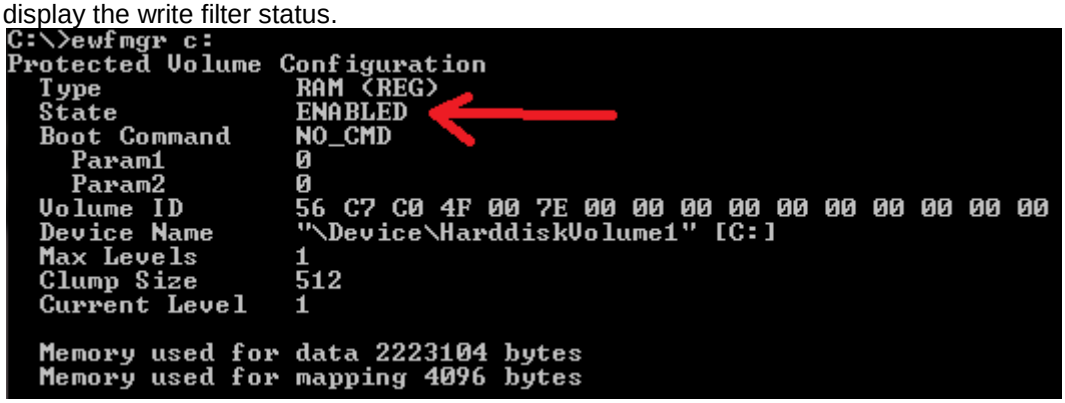


**Figure 3 Write Filter Enabled After Reboot**

### 1.2 Discard Changes and Disable After Next Reboot.

Typing "**`ewfmgr c: -disable`**" will disable the write filter AFTER NEXT REBOOT and NOT commit any changes to CompactFLASH.

```
C:\>ewfmgr c: -disable
*** Disabling overlay

Protected Volume Configuration
  Type             RAM (REG)
  State            ENABLED
  Boot Command     DISABLE
    Param1         0
    Param2         0
  Volume ID        56 C7 C0 4F 00 7E 00 00 00 00 00 00 00 00 00 00
  Device Name      "\Device\HarddiskVolume1" [C:]
  Max Levels       1
  Clump Size       512
  Current Level    1

  Memory used for data 2920448 bytes
  Memory used for mapping 4096 bytes
```

**Figure 4 Write Filter Will be Disabled Next Boot**

### 1.3 Commit Changes and Disable After Next Reboot.

Typing "**`ewfmgr c: -commitanddisable`**" will commit changes to CompactFLASH and disable the write filter next reboot.

```
C:\>ewfmgr c: -commitanddisable
*** Committing data and disabling overlay

Protected Volume Configuration
  Type             RAM (REG)
  State            ENABLED
  Boot Command     DISABLE
    Param1         1
    Param2         0
  Volume ID        56 C7 C0 4F 00 7E 00 00 00 00 00 00 00 00 00 00
  Device Name      "\Device\HarddiskVolume1" [C:]
  Max Levels       1
  Clump Size       512
  Current Level    1

  Memory used for data 2276864 bytes
  Memory used for mapping 4096 bytes
```

**Figure 5 Committing and Disabling Write Filter**

### 1.4 Commit Changes and Disable Immediately.

Typing "**ewfmgr c: -commitanddisable -live**" will commit changes to CompactFLASH immediately and disable the write filter without requiring a reboot.

```
C:\>ewfmgr c: -commitanddisable -live
*** Committing data and disabling overlay (live)

Protected Volume Configuration
  Type             RAM (REG)
  State            DISABLED
  Boot Command     NO_CMD
    Param1         0
    Param2         0
  Volume ID        56 C7 C0 4F 00 7E 00 00 00 00 00 00 00 00 00 00
  Device Name      "\Device\HarddiskVolume1" [C:]
  Max Levels       1
  Clump Size       512
  Current Level    N/A

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

**Figure 6 Committed Live and Disabled**

### 1.5 Commit Changes Next Reboot.

Typing "**ewfmgr c: -commit**" will commit any changes to CompactFLASH next reboot. After reboot the write filter will still be enabled.

```
C:\>ewfmgr c: -commit
*** Committing overlay to the protected volume.

Protected Volume Configuration
  Type             RAM (REG)
  State            ENABLED
  Boot Command     COMMIT
    Param1         0
    Param2         0
  Volume ID        56 C7 C0 4F 00 7E 00 00 00 00 00 00 00 00 00 00
  Device Name      "\Device\HarddiskVolume1" [C:]
  Max Levels       1
  Clump Size       512
  Current Level    1

  Memory used for data 2268160 bytes
  Memory used for mapping 4096 bytes
```

**Figure 7 Changes Will Committed Next Reboot**

## 2 Monitoring the Write Filter.

As data is written to disk system RAM is consumed and if excessive writing is allowed to go unchecked the system will possibly run out of memory.

The amount of RAM used by the write filter can be determined by Typing "**ewfmgr c:**"

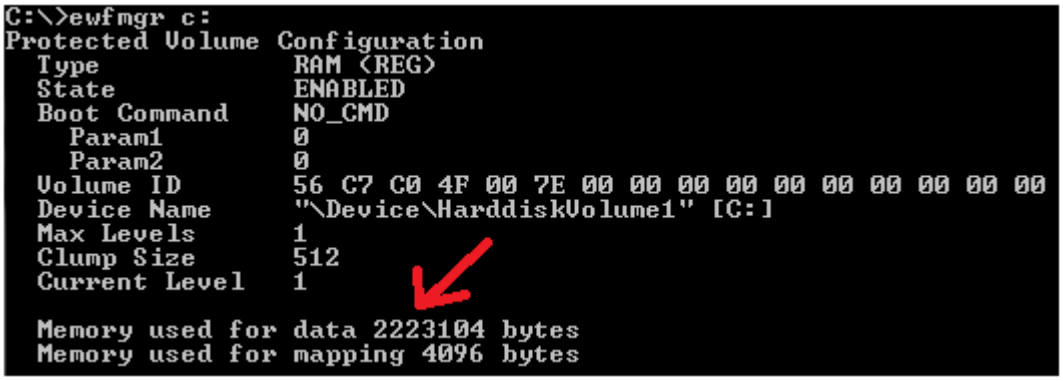


**Figure 8 RAM Used by Write Filter**

It is difficult to quote an acceptable value for memory used, as it depends on how application software is using the system. As a general rule of thumb in a typical XP Embedded system the value will increase from typically 100,000 bytes at start up to a couple of Mbyte once the system is running eg 2-4Mbyte is typical. It should then stay at fixed at this value.

Any constant increase of RAM used over time should be investigated, to determine which process is writing to disk, or else the system will eventually run out of RAM.