

Microsoft Windows Embedded Standard 2009

Syslogic User's Manual

Document Ordercode: IPC/WINESTD09-7A



Revision	Datum	Author	Modification
1.0	29.06.2011	Roger Newbould	Based on IPC/WINXPE-6A doc, v1.6-
1.1	30.10.2013	Roger Newbould	Errata USB device removal
1.2	16.12.2013	Roger Newbould	Support for IPC71 added

Content

1	General Remarks	5
2	Introduction	5
2.1.	Supported Hardware	6
3	Getting Started	6
3.1.	Booting WES2009	6
3.2.	User and Password Settings	7
3.3.	Drive Write Protection	7
4	Windows Embedded Standard 2009 Image Features	8
4.1.	Features included	9
4.2.	Features NOT included	9
4.3.	Default Configuration	9
4.4.	Tools	10
4.5.	Touch Driver	11
4.6.	Graphics Driver for IPC/COMPACT7 and IPC/ML71-Systems	11
4.6.1.	Intel Embedded Graphics Driver	11
4.6.2.	Intel Embedded Media and Graphics Driver	12
4.7.	Graphics Driver for IPC/COMPACT6-Systems	12
4.8.	Ethernet Drivers for IPC/COMPACT6-Systems	12
4.8.1.	Intel i82551	12
4.8.2.	Davicom DM9102	12
5	File Based Write Filter	12
5.1.	Overview	12
5.2.	Configuring EWF – EWF Manager Tool	14
5.2.1.	FBWF Manager	14
5.2.2.	FBWF Manager Command Line Syntax	14
5.3.	Configuring FBWF – FBWF Manager Tool	17
5.3.1.	FBWF Manager	17
5.3.2.	FBWF Manager Command Line Syntax	18
5.4.	Committing files and adding exclusions	21
5.5.	Compact flash drive space	22
5.6.	Wear Leveling Algorithms explained	23
5.7.	Endurance Calculation	25
6	Power Failure	26

6.1.	Overview	26
6.2.	Power Failure - FBWF enabled	26
6.3.	Power Failure – FBWF disabled	26
7	System Errors - Blue Screens	27
7.1.	Overview	27
7.2.	Enable Auto Reboot	27
8	Remote Debugging	28
8.1.	Connect and Debug	28
9	Installing WES2009	30
9.1.	Overview	30
9.2.	Prepare the CompactFlash for File Copy	30
9.3.	Copy the WES2009 files to the Compact Flash	32
9.4.	Post FBA setup	33
10	WinIo Driver Library	33
10.1.	WinIo Library Overview	33
10.2.	NETIPC extensions to WinIO	34
10.3.	Using WinIo in a Visual C++ application	34
10.4.	TestProg Sample Application	35
10.5.	WinIO End User License Agreement	35
11	Errata	36
11.1.	USB Storage Device Removal	36
11.1.1.	Description	36
11.1.2.	Workaround	36
12	References	37
13	Contact	38
List of figures		
Figure 2 –	Folder structure	11
Figure 3 –	fbwfmgr.exe utility	18
Figure 4 –	Disk C: properties	22
Figure 5 –	No wear leveling	23
Figure 6 –	Dynamic wear leveling	23
Figure 7 –	Static wear leveling	23
Figure 8 –	Filemon output	24
Figure 9 -	Auto Reboot feature	27
Figure 10 -	Remote Application Debugging	28
Figure 11 –	Remote Debugging Setup	29
Figure 12 –	Windows explorer	30

Figure 13 – Format	31
Figure 14 – WES2009 filecopy setup	32
Figure 15 – Eject CompactFlash	33

1 General Remarks

The content and presentation of this document has been carefully checked. No responsibility is accepted for any errors or omissions in the documentation. Note that the documentation for the products is constantly revised and improved. The right to change this documentation at any time without notice is therefore reserved.

Syslogic is grateful for any help referring to errors or for suggestions for improvements.
The following registered trademarks are used:

IBM-PC, PC/AT, PS/2	trademarks of IBM Corporation
Windows XP Embedded, XPe	trademark of Microsoft Corporation
Windows Embedded Standard	trademark of Microsoft Corporation
WES2009	trademark of Microsoft Corporation
FBWF	trademark of Microsoft Corporation
IPC/NETIPC	trademark of Syslogic Datentechnik AG
IPC/NETSBC	trademark of Syslogic Datentechnik AG
IPC/COMPACT7	trademark of Syslogic Datentechnik AG
FILEMON	trademark of Sysinternals
WinIO	trademark of Yariv Kaplan

2 Introduction

Windows Embedded Standard 2009 is formely known as Windows XP Embedded and has SP3 integrated.. It is based on the same kernel and is the componentized version of Microsoft Windows XP Professional. In fact, both Operating Systems are based on the same binaries.

WES2009 is enhanced with special tools to target the needs of embedded systems that run continuously. While a desktop OS like XP Professional is optimized for maximum user interaction, an embedded OS is the opposite. Embedded systems often run in environments where few user interactions are desired. Embedded systems often supervise safety critical facilities and thus need to be very robust, reliable and autonomous.

This user manual covers some tools and techniques that help to reduce possible software/hardware breakdowns concerning the operation of Windows Embedded Standard on a IPC in an industrial environment. It details some embedded features that can be configured. Basic knowledge of Windows operating systems is required.

2.1. Supported Hardware

This release of WES2009 supports the following systems from Syslogic:

Syslogic produc	Product features
IPC/COMPACT6-xxx	AMD Geode LX800 based IPC
IPC/COMPACT7L1-xxx	Intel Atom Z510 based IPC with 1GB DRAM
IPC/COMPACT7M1-xxx	Intel Atom Z530 based IPC with 1GB DRAM
IPC/COMPACT7M2-xxx	Intel Atom Z530 based IPC with 2GB DRAM
IPC/SL71F16-xxx	Intel Atom E680T based IPC with 1GB DRAM
IPC/SL71G16-xxx	Intel Atom E680T based IPC with 2GB DRAM
IPC/ML71F16-xxx	Intel Atom E680T based IPC with 1GB DRAM
IPC/ML71G16-xxx	Intel Atom E680T based IPC with 2GB DRAM

Table 1 – Supported CPU boards

3 Getting Started

This chapter will give you a “quick start” on how to get WES2009 running on a Intel Atom based system from Syslogic. The same procedure applies for all Systems.

3.1. Booting WES2009

If you don't have a CompactFlash card with a fully installed WES2009 image please read chapter 9 on how to install the system files.

Open your industrial PC (IPC) and plug the Compact flash card into the corresponding socket. Close the IPC and connect a USB keyboard, mouse and a monitor. Power the IPC and Windows Embedded Standard should start loading. If the IPC doesn't boot from the CompactFlash (drive C) then the appropriate setting has to be change in the BIOS setup.

CMOS Setup:

Boot 1st: Drive C:
Ide 0: 3 = AUTOCONFIG, LBA

Hit “ESC” to leave the configuration screen and select “*Write to CMOS and Exit*” (confirm with Y or Z key). Now, the system should boot into Windows XP Embedded. You do not always have to change these CMOS settings when booting WES2009, as these CMOS settings

are stored permanently if (and only if) the battery is inserted and operational. If you change the battery, you must redo the correct settings.

3.2. User and Password Settings

The Syslogic WES2009 image is configured for Autologon, i.e. if the system starts, there's no login screen showing up and the default user "Administrator" is automatically logged on.

User and Password:

WES2009 Autologon User:	Administrator
WES2009 Autologon Password:	netipc

3.3. Drive Write Protection

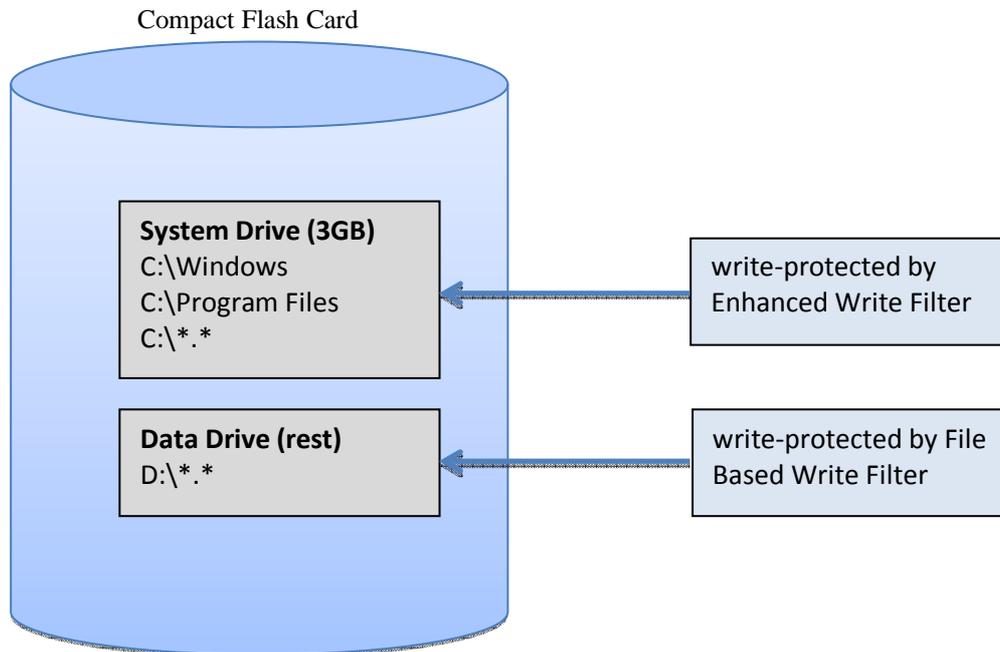
The major drawback of Compact flash cards is their limited write cycles of typically 100'000 operations per block before breakdown. Thus it's extremely important to protect the Compact Flash card in terms of write accesses.

This protection is achieved through the Enhanced and File Based Write Filter (EWF and FBWF) shipped with Windows Embedded Standard 2009.

Enhanced Write Filter is a Block Level Filter, redirecting each write of a block to a RAM Layer, thus can protect all writes to an underlying disk. EWF does support all features of the desired File System and the Operation System like NTFS and Hibernate / HORM. EWF is the best choice to protect the system drive containing the operation system. Because EWF is working on a block base, it is only possible to protect the whole drive.

File Based Write Filter is a File Level Filter, redirecting each write of a file to a RAM Layer. Because FBWF is working on per File or per Directory base, FBWF allows you to make exceptions of folders or files which should not be protected. This allows you to protect a whole drive but still write all files down to the disk for some dedicated folders like a Configuration Folder.

To reach the best protection and supporting most features of the operation system, the Compact Flash Card is splitted in two partitions. The system partition is protected by EWF and the data partition is protected by FBWF.



Per factory default the FBWF is disabled. So if you boot into WES2009 and change some system settings, these settings will persist between reboots.

Note:

If EWF is enabled, system and drive C: settings are write-protected. For FBWF you have first to add the desired volumes (D-Drive) to the FBWF. If y volume is added, ALL files and folders of the drive are protected. Add exceptions to define files and folders as write through.

If you create a file C:\Windows\myfile.txt with EWF enabled, this file will not exist after the next reboot. We heavily encourage you to read the following sections, especially the section on the “Enhanced Write Filter” and File Based Write Filter”, to get further information on how to enable/disable and use the EWF and FBWF features.

4 Windows Embedded Standard 2009 Image Features

The software included in our Windows Embedded Standard 2009 image is a subset of the software shipping with Windows XP Desktop edition. From 14'000 selectable packages, there are around 1500 currently included into our standard WES2009 configuration.

So do not expect to have the full Desktop XP Pro feature set included in our standard image. But be also aware of the fact that you can install your own missing drivers and software packages at will like on a regular Windows XP system.

Below you will find a list of features that are included/not included in our standard configuration. This list is not complete, it may only give you an overview of the basic software bundles that are shipped with our WES2009 standard image.

4.1. Features included

The following list is not complete.

- Windows XP user interface, full desktop with taskbar and start menu
- Command shell
- Control panel
- WMI Scripting Interface
- Internet Explorer 7
- Windows file explorer
- Microsoft .NET2.0 SP2, .NET3.0 SP2, .NET3.5 SP1, .NET4.0 Client Profile and Extended
- Microsoft installer
- Windows media player 11
- Remote desktop connection, Hyperterminal
- Remote folder share, drive mapping, basic networking utilities
- Drivers for board hardware (mouse & keyboard PS/2 & USB, flash-memory-stick, video).
- IR touch screen driver installation executable

4.2. Features NOT included

- Java Virtual Machine
- Most user applications like Games, etc.
- Language Support for Languages other than English & German
- Drivers for additional hardware devices

4.3. Default Configuration

- Firewall disabled
- Autologin enabled
- Just user Administrator defined
- EWF and FBWF disabled
- English language support
- DHCP enabled
- Computer name: "Wes09"

Please consult the WES2009 build logfile IPC_WINESTD09_7A_vnnn.log or IPC_WINESTD09_71A_vnnn.log from the product DVD-ROM to see the full list of packages which are included into the WES2009 release.

4.4. Tools

A folder named “Utilities” is placed on the desktop and it contains a few useful tools:

- Autologon: tool for enabling and disabling the logon screen
- Imagemaster: a tool to backup and restore a whole CompactFlash
- Touch: IR Touch installer for display systems
- WinIo: modified WinIo library with functions for accessing peripherals
- Original_WinIo: original WinIo library for accessing I/O and memory resources under Windows

4.5. Touch Driver

The touch driver installer is located in a folder on the Desktop of your system.

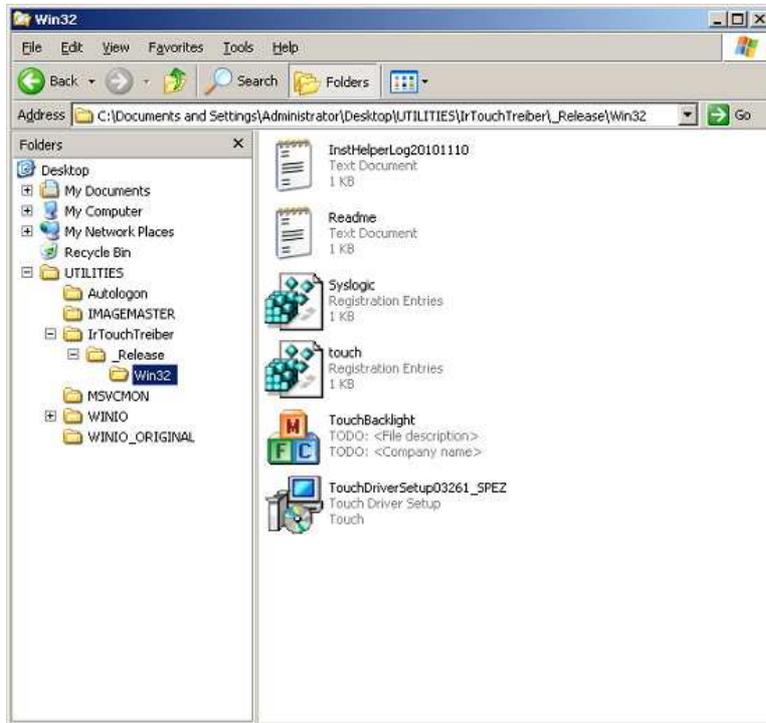


Figure 1 – Folder structure

- TouchDriverSetup03261_SPEZ.exe installation executable
- Syslogic.reg registry patch
- TouchBacklight.exe tool to change backlight intensity

After the driver is installed successfully, double click onto *Syslogic.reg* to automatically modify certain registry keys.

4.6. Graphics Driver for IPC/COMPACT7 and IPC/ML71-Systems

Intel is modifying its graphic driver constantly. It's suggested to visit Intel's website from time to time for new driver versions. Intel has developed two driver architectures: IEGD and EMGD. It's recommended to use the later driver.

4.6.1. Intel Embedded Graphics Driver

IEGD specifically targets the needs of embedded platform developers and it supports a large variety of embedded processor and chipsets.

Check the link below for the newest version:

<http://edc.intel.com/Software/Downloads/IEGD/>

4.6.2. Intel Embedded Media and Graphics Driver

The new Intel EMGD is based on a new optimized architecture, offering long-life production commitment and enables new graphic features. The Intel EMGD is developed specifically for the Intel Atom Z5xx and E6xx processor series, including its chipsets.

Check the link below for the newest version:

<http://edc.intel.com/Software/Downloads/EMGD/>

4.7. Graphics Driver for IPC/COMPACT6-Systems

For AMD Geode LX800 based systems the XPe graphics driver is installed (v3.00.02). According to AMD no new versions are planned.

4.8. Ethernet Drivers for IPC/COMPACT6-Systems

Newer IPC/COMPACT6 systems have two DM9102 ethernet controller from Davicom. For backward compatibility purposes the driver for the Intel i82551 ethernet is also installed. During start-up Windows detects the controller and loads the driver automatically.

4.8.1. Intel i82551

Version 8.0.13.0 of the i82551 driver is installed.

4.8.2. Davicom DM9102

Version v1.55 of the DM92102 XP driver from Davicom is installed.

5 File Based Write Filter

5.1. Overview

Compact flash cards are a very popular storage for data because of their low price, small form factor and robustness (e.g. no moving parts, high temperature operation). The major drawback of compact flash cards is their limited write robustness of typically 100'000 operations per block before breakdown. The Compact flash card is not wear resistant against excessive write cycles.

Thus it's extremely important to protect the Compact Flash card in terms of write accesses. The Enhanced Write Filter "EWF" and the File Based Write Filter "FBWF" shipped with Windows Embedded Standard protects the flash card. If the EWF/FBWF is enabled, every data written to drives are actually redirected to a RAM layer, thus the Compact Flash is protected against writes (see figure 1).

Important:

When using Compact flash with Windows WES2009, the EWF must be enabled.

Of course, this setup results in a stateless system. After a reboot, all data previously written will be lost. If EWF/FBWF is enabled, the Windows WES2009 OS is also secured against accidental system file damages (i.e. if an application deleted some important system files).

If you want to store application data permanently, you can use an unprotected folder on drive D:\ or you can disable the EWF/FBWF.

5.2. Configuring EWF – EWF Manager Tool

WES2009 includes the “FBWF Manager Console Application”, FBWFMGR.EXE, which controls the FBWF configuration. One can query the state of the FBWF feature by simply executing the command *fbwfmgr* in a Windows shell.

5.2.1. FBWF Manager

Source: <http://msdn.microsoft.com/en-us/library/aa940817%28v=WinEmbedded.5%29.aspx>

The FBWF Manager is a command line tool for embedded developers to quickly integrate and prototype the FBWF. FBWF Manager is intended primarily for design time use.

5.2.2. 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 [path] | /removeexclusion [path] |  
/setthreshold [threshold] | /setcompression [1|0] | /setpreallocation [1|0]  
/commit [volumename] [filepath] /restore [volumename] [filepath] ]
```

The following table describes the command line switches.

Switch	Description
<i>displayconfig</i>	Displays all configuration information for the write filter including protected volumes list, overlay configuration and write through paths. The command returns: State—Indicating current filter state (enable or disable) and state for next boot. Protected Volumes—List of protected volumes including the current and next boot state. Compression—Current and next boot state for cache compression.

	<p>Threshold—Current and next boot values for the overlay cache threshold.</p> <p>Write Through Paths—Displays a complete list of active and next boot write through paths.</p> <p>Pre-allocation Status—Displays current and next boot status for cache pre-allocation.</p>
<i>overlaydetail</i>	<p>Displays detail on the current overlay contents for all protected volumes. The command returns:</p> <p>Contents—Files and folders currently in the overlay for all protected volumes including sizes (size of data in overlay) and open file handles.</p> <p>Memory Usage—Total amount of memory being consumed by the overlay.</p>
<i>enable</i>	Enables the write filter on the next restart.
<i>disable</i>	Disables the write filter on the next restart.
<i>addvolume</i>	Adds a volume to the protected volume list for next boot.
<i>removevolume</i>	Removes a volume from the protected volume list for next boot.
<i>addexclusion</i>	Adds a write through path to the exclusion list for next boot.
<i>removeexclusion</i>	Removes a write through path from the exclusion list for next boot.
<i>setthreshold</i>	Sets the overlay threshold value for next boot.
<i>setcompression</i>	Sets overlay compression as enabled (1) or disabled (0) for next boot.
<i>setpreallocation</i>	Sets cache pre-allocation as enabled (1) or disabled (0) for next boot.
<i>commit</i>	<p>Commits the changes made to the file to the underlying media.</p> <p>The volume name can either be a case-insensitive volume device name (for example, "\\Device\\HarddiskVolume1"), or a drive letter (for example, "C:" or "D:").</p> <p>Note that the name is not the volume label that Windows Explorer displays before the drive letter.</p>

	<p>The file path must be an absolute path starting with "\".</p> <p>Note that the volume must currently be protected. Otherwise, the error message "The system cannot find the drive specified" is displayed.</p>
<i>restore</i>	<p>Discards the changes made to the file, that is, restores the files to its original contents from the underlying media.</p> <p>The volume name can either be a case-insensitive volume device name (for example, "\\Device\\HarddiskVolume1"), or a drive letter (for example, "C:" or "D:").</p> <p>Note that the name is not the volume label that Windows Explorer displays before the drive letter.</p> <p>The file path must be an absolute path starting with "\". It must be a file. It is acceptable that the file was deleted, in which case it is recovered.</p> <p>Note that the volume must currently be protected. Otherwise, the error message "The system cannot find the drive specified" is displayed.</p>
<i>?</i>	Displays usage and help.
<i>help / [switch]</i>	Displays help information for a specific FBWF Manager switch.

If no switch is provided the FBWF Manager displays all the configuration information, just like the *DisplayConfig* switch.

The following table describes the input parameters.

Input field	Meaning
<i>volumename</i>	Full path to a volume
<i>1</i>	Remove exclusion list
<i>0</i>	Preserve exclusion list
<i>path</i>	<p>Full file or directory path, including the drive letter.</p> <p>Please note that file names are passed to fbwfmgr as a command line argument, which means backslashes and double</p>

	quotes are interpreted differently. For example, <code>\"file name"</code> becomes simply <code>"file name"</code> because the first backslash acts as an escape character. To get <code>\"file name"</code> , specify <code>\\\"file name"</code> on the command line.
<i>threshold</i>	Overlay threshold in MB

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

FBWF does not resolve substituted paths, use complete file names instead.

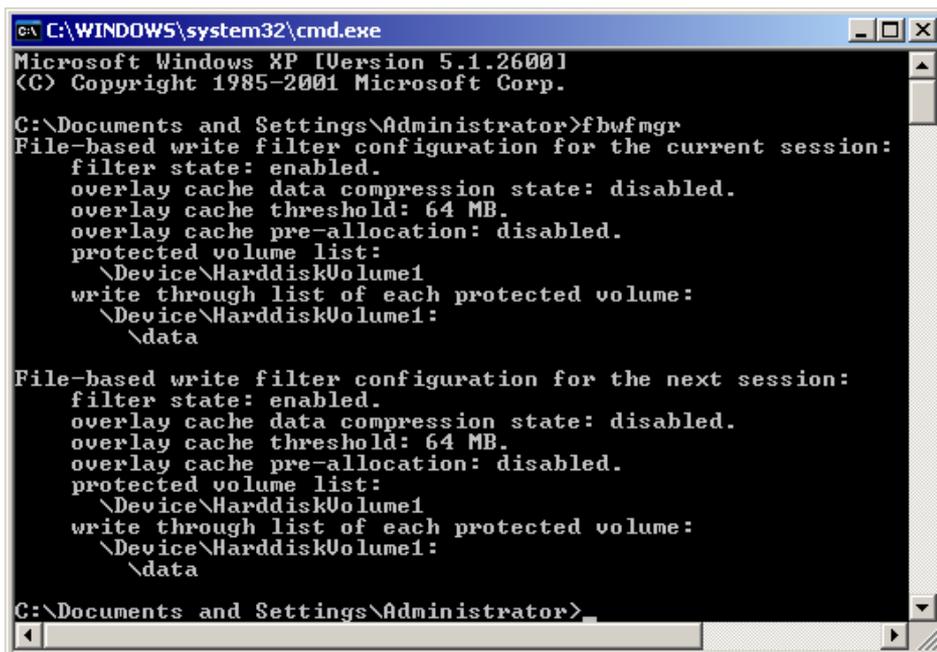
5.3. Configuring FBWF – FBWF Manager Tool

WES2009 includes the “FBWF Manager Console Application”, `FBWFMGR.EXE`, which controls the FBWF configuration. One can query the state of the FBWF feature by simply executing the command `fbwfmgr` in a Windows shell.

5.3.1. FBWF Manager

Source: <http://msdn.microsoft.com/en-us/library/aa940817%28v=WinEmbedded.5%29.aspx>

The FBWF Manager is a command line tool for embedded developers to quickly integrate and prototype the FBWF. FBWF Manager is intended primarily for design time use.



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>fbwfmgr
File-based write filter configuration for the current session:
  filter state: enabled.
  overlay cache data compression state: disabled.
  overlay cache threshold: 64 MB.
  overlay cache pre-allocation: disabled.
  protected volume list:
    \Device\HarddiskVolume1
  write through list of each protected volume:
    \Device\HarddiskVolume1:
      \data

File-based write filter configuration for the next session:
  filter state: enabled.
  overlay cache data compression state: disabled.
  overlay cache threshold: 64 MB.
  overlay cache pre-allocation: disabled.
  protected volume list:
    \Device\HarddiskVolume1
  write through list of each protected volume:
    \Device\HarddiskVolume1:
      \data

C:\Documents and Settings\Administrator>
```

Figure 2 – fbwfmgr.exe utility

To enable or disable the FBWF RAM overlay of drive C:, one must open a shell console in Windows XP Embedded and use the following command:

```
C:\> fbwfmgr /enable  
C:\> fbwfmgr /disable
```

This will install or remove the RAM overlay of partition drive C:. These operations require a complete system reboot to effect the changes. Don't expect the FBWF being enabled/disabled just by executing the appropriate command. A power cycle is mandatory.

Important:
Each configuration change of the FBWF needs a complete system reboot until the change is effective.

5.3.2. 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 [path] | /removeexclusion [path] |  
/setthreshold [threshold] | /setcompression [1|0] | /setpreallocation [1|0]  
/commit [volumename] [filepath] /restore [volumename] [filepath] ]
```

The following table describes the command line switches.

Switch	Description
<i>displayconfig</i>	Displays all configuration information for the write filter including protected volumes list, overlay configuration and write through paths. The command returns: State—Indicating current filter state (enable or disable) and state for next boot. Protected Volumes—List of protected volumes including the

	<p>current and next boot state.</p> <p>Compression—Current and next boot state for cache compression.</p> <p>Threshold—Current and next boot values for the overlay cache threshold.</p> <p>Write Through Paths—Displays a complete list of active and next boot write through paths.</p> <p>Pre-allocation Status—Displays current and next boot status for cache pre-allocation.</p>
<i>overlaydetail</i>	<p>Displays detail on the current overlay contents for all protected volumes. The command returns:</p> <p>Contents—Files and folders currently in the overlay for all protected volumes including sizes (size of data in overlay) and open file handles.</p> <p>Memory Usage—Total amount of memory being consumed by the overlay.</p>
<i>enable</i>	Enables the write filter on the next restart.
<i>disable</i>	Disables the write filter on the next restart.
<i>addvolume</i>	Adds a volume to the protected volume list for next boot.
<i>removevolume</i>	Removes a volume from the protected volume list for next boot.
<i>addexclusion</i>	Adds a write through path to the exclusion list for next boot.
<i>removeexclusion</i>	Removes a write through path from the exclusion list for next boot.
<i>setthreshold</i>	Sets the overlay threshold value for next boot.
<i>setcompression</i>	Sets overlay compression as enabled (1) or disabled (0) for next boot.
<i>setpreallocation</i>	Sets cache pre-allocation as enabled (1) or disabled (0) for next boot.
<i>commit</i>	<p>Commits the changes made to the file to the underlying media.</p> <p>The volume name can either be a case-insensitive volume device name (for example, "\\Device\\HarddiskVolume1"),</p>

	<p>or a drive letter (for example, "C:" or "D:").</p> <p>Note that the name is not the volume label that Windows Explorer displays before the drive letter.</p> <p>The file path must be an absolute path starting with "\\".</p> <p>Note that the volume must currently be protected. Otherwise, the error message "The system cannot find the drive specified" is displayed.</p>
<i>restore</i>	<p>Discards the changes made to the file, that is, restores the files to its original contents from the underlying media.</p> <p>The volume name can either be a case-insensitive volume device name (for example, "\\Device\\HarddiskVolume1"), or a drive letter (for example, "C:" or "D:").</p> <p>Note that the name is not the volume label that Windows Explorer displays before the drive letter.</p> <p>The file path must be an absolute path starting with "\\". It must be a file. It is acceptable that the file was deleted, in which case it is recovered.</p> <p>Note that the volume must currently be protected. Otherwise, the error message "The system cannot find the drive specified" is displayed.</p>
?	Displays usage and help.
<i>help / [switch]</i>	Displays help information for a specific FBWF Manager switch.

If no switch is provided the FBWF Manager displays all the configuration information, just like the *DisplayConfig* switch.

The following table describes the input parameters.

Input field	Meaning
<i>volumename</i>	Full path to a volume

<i>1</i>	Remove exclusion list
<i>0</i>	Preserve exclusion list
<i>path</i>	Full file or directory path, including the drive letter. Please note that file names are passed to fbwfmgr as a command line argument, which means backslashes and double quotes are interpreted differently. For example, <code>\"file name"</code> becomes simply <code>"file name"</code> because the first backslash acts as an escape character. To get <code>\"file name"</code> , specify <code>\\\"file name"</code> on the command line.
<i>threshold</i>	Overlay threshold in MB

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

FBWF does not resolve substituted paths, use complete file names instead.

5.4. Committing files and adding exclusions

It is possible to commit a file from the protected region to the Compact flash card. As an example, we will create a file on the protected C:\ root directory called myfile.txt and will commit it to the compact flash card.

```
C:\> echo hello > c:\myfile.txt
C:\> fbwfmgr /commit c: \myfile.txt      (note the space between c: and \myfile.txt)
```

After the next reboot, myfile.txt will still exist.

If you like to add some folders that are permanently writable, you could make use of the "addexclusion" argument of FBWF. The following example will change the directory test state from protected to unprotected.

```
C:\> mkdir c:\test
C:\> fbwfmgr /addexclusion c: \test      (note the space between c: and \myfile.txt)
```

Again, reboot the system and note that the directory test still exists.

For further usage examples of the FBWF manager tool, please consult the document *FBWF.pdf* from the DVD-ROM.

5.5. Compact flash drive space

When opening the drive C: properties page with FBWF enabled, one thing could lead to confusion: Although you have a 1GByte Compact flash card inserted, the properties page displays a disk capacity of ~490MBytes with only ~60MBytes free space left.

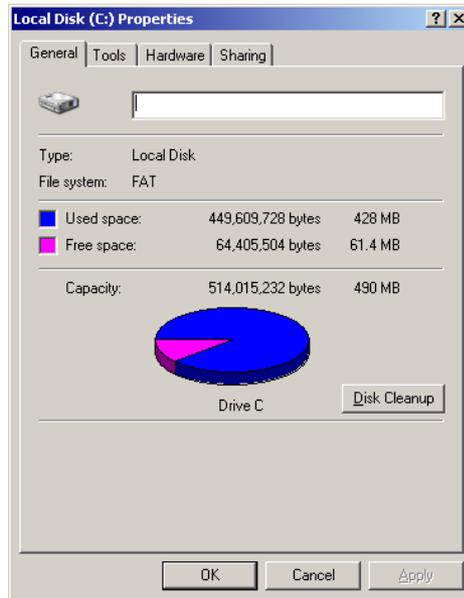


Figure 3 – Disk C: properties

This might be confusing but it is not a system fault. With FBWF enabled, the write cycles to drive C: are filtered into RAM layer which is limited to 64MBytes. So if you download a file of size 10MBytes from the Internet to the Desktop, the free space would decrease 10MBytes. If you delete that file or if you reboot, the 10MBytes would be free again.

With FBWF disabled, you would see the “real” free space on the Compact flash card which would be around 510 MBytes.

sCompact Flash Cards

5.6. Wear Leveling Algorithms explained

Let's assume a Compact flash card that has 6 blocks, where 3 blocks are filled with system data (i.e. operating system files on drive C:) that were written once. This might be a typical situation where the FBWF filter is enabled on the first 3 blocks (see section 5). The first 3 system file blocks are protected by the FBWF, where the remaining 3 blocks are unprotected on the Compact flash card. We further assume that a user application is writing a single block of data to the same logical block address of the Compact flash card for 99 times.

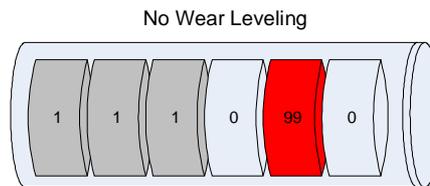


Figure 4 – No wear leveling

If the Compact flash card does not implement wear-leveling, then all 99 writes happen on the same block and the flash is destroyed soon (see figure 2 – the numbers represent the write access count of each block).

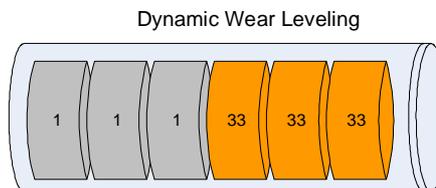


Figure 5 – Dynamic wear leveling

If the Compact flash does implement a dynamic wear leveling algorithm, then the write accesses are evenly distributed across the free blocks, if the application always writes to the same file. This leads to much better endurance of the flash (see figure 3).

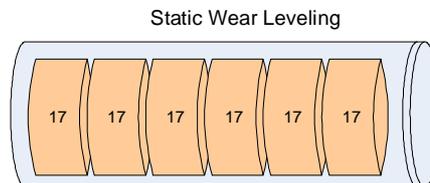


Figure 6 – Static wear leveling

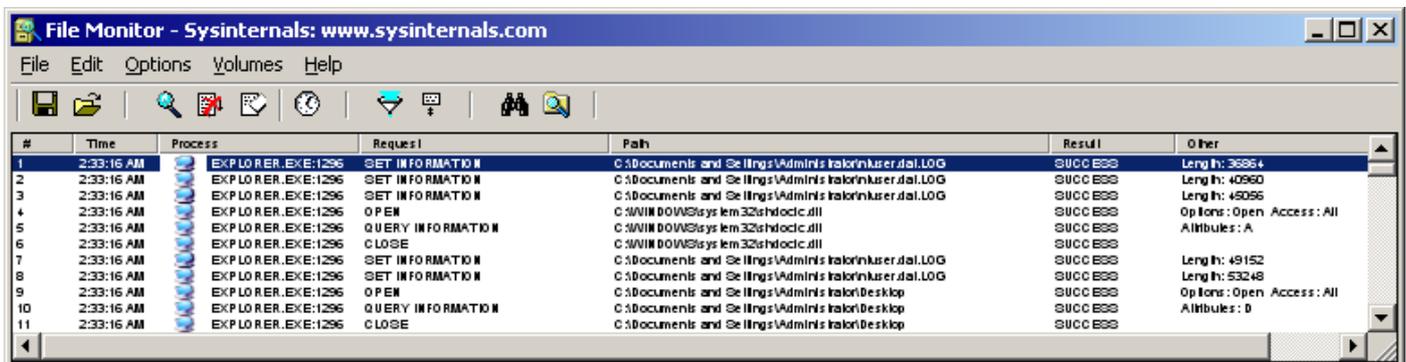
Maximum endurance is achieved with a Compact flash card that implements Static Wear-Leveling. This algorithm spreads the write cycles over the entire flash capacity, including the blocks that are protected by FBWF.

You might wonder where the FBWF protection has gone with static wear-leveling. On the first glance, the FBWF features seem useless with a Compact flash card that implements static wear-leveling.

Let me explain this by introducing two different FBWF protection goals:

- flash protection against user program write accesses
- flash protection against WES2009 system activity write accesses

If you use a file monitor tool (e.g. FILEMON from www.sysinternals.com), you will notice a lot of write activity caused by the WES2009 system. E.g. if you open the internet explorer application on the desktop, the window properties will be stored in the file user.dat as shown in the following figure.



#	Time	Process	Request	Path	Result	Other
1	2:33:16 AM	EXPLORER.EXE:1296	SET INFORMATION	C:\Documents and Settings\Administrator\user.dat.LOG	SUCCESS	Length: 3596+
2	2:33:16 AM	EXPLORER.EXE:1296	SET INFORMATION	C:\Documents and Settings\Administrator\user.dat.LOG	SUCCESS	Length: 40960
3	2:33:16 AM	EXPLORER.EXE:1296	SET INFORMATION	C:\Documents and Settings\Administrator\user.dat.LOG	SUCCESS	Length: 4096
4	2:33:16 AM	EXPLORER.EXE:1296	OPEN	C:\WINDOWS\system32\shdoc.dll	SUCCESS	Options: Open Access: All Attributes: A
5	2:33:16 AM	EXPLORER.EXE:1296	QUERY INFORMATION	C:\WINDOWS\system32\shdoc.dll	SUCCESS	
6	2:33:16 AM	EXPLORER.EXE:1296	CLOSE	C:\WINDOWS\system32\shdoc.dll	SUCCESS	
7	2:33:16 AM	EXPLORER.EXE:1296	SET INFORMATION	C:\Documents and Settings\Administrator\user.dat.LOG	SUCCESS	Length: 49152
8	2:33:16 AM	EXPLORER.EXE:1296	SET INFORMATION	C:\Documents and Settings\Administrator\user.dat.LOG	SUCCESS	Length: 53248
9	2:33:16 AM	EXPLORER.EXE:1296	OPEN	C:\Documents and Settings\Administrator\Desktop	SUCCESS	Options: Open Access: All
10	2:33:16 AM	EXPLORER.EXE:1296	QUERY INFORMATION	C:\Documents and Settings\Administrator\Desktop	SUCCESS	Attributes: 0
11	2:33:16 AM	EXPLORER.EXE:1296	CLOSE	C:\Documents and Settings\Administrator\Desktop	SUCCESS	

Figure 7 – Filemon output

The main reason for FBWF is to protect the Compact flash card from suffering damage by these WES2009 internal system activities. As the programmer of the embedded system often has total control over his application and the amount of write accesses to an unprotected file, the flash protection against user program write accesses is not the main concern/problem.

In fact, Compact flash cards implementing static wear leveling have so much better endurance concerning write accesses, that the advantage might overwhelm the fact that the system files on a Compact flash card implementing dynamic wear-leveling are undestroyable.

Big Picture:

dynamic wear-leveling and FBWF:

- WES2009 system blocks can never be damaged
- application can not write much user data to unprotected files

static wear-leveling and FBWF:

- WES2009 system file blocks can suffer damage when the flash wears out
- application can write much more data to unprotected files before damage can occur.

Syslogic recommends the use of Compact Flash cards that implement Static Wear-Leveling like the SiliconDrive™ Compact flash card from Silicon Systems.

5.7. Endurance Calculation

To get an idea of how long a Compact flash card with a Static Wear-Leveling algorithm can be used, the following formula could be helpful. This formula assumes a Compact Flash with a flash block endurance of 100'000 write cycles.

$$Years = \frac{CardSize}{WriteRate \cdot 310}$$

Data size unit is *MBytes* and *WriteRate* is in *MBytes per Second*. For an application with a CompactFlash capacity of 1024 MBytes and a data file of 0.1MBytes that is written 10 times per second, this formula results in a card endurance of 3.3 years.

With SiliconSystems Compact flash cards, the max. write transfer rate is 6MBytes/s, so the minimal endurance for the 1024MByte card is 0.5 years, assuming full write workload.

Please note that the formula above is only an approximation and only correct for Compact flash drives that implement Static Wear-Leveling. Drives that use other algorithms like Dynamic Wear-Leveling or no Wear-Leveling have much less endurance.

Note: The CompactFlash block endurance is assumed as 100k write cycles in the formula above. For other block endurance specifications, the formula can be simply scaled linear with the block endurance enhancement factor.

6 Power Failure

6.1. Overview

Writing a file (consisting of multiple sectors/blocks) to a Compact Flash card is not an atomic operation, so there could be a possible file system corruption upon a power loss. FAT32 file systems typically used under Windows XP can be corrupted. NTFS file systems are less sensitive to corruption, as they implement a journal that keeps track of the directory structure and recovers the inconsistencies while booting. But it is also possible to corrupt an NTFS file system.

6.2. Power Failure - FBWF enabled

When enabling the Enhanced Write Filter according to section 0, there is no danger in terms of system file corruption because the system drive partition is write protected. So the system will always boot when the power is switched on again.

Good to know:

FBWF also protects the system files from corruption in case of a power failure.

6.3. Power Failure - FBWF disabled

If you intend not to use FBWF, then an NTFS formatted system drive would be the first choice, although such a setup is not 100% tolerant to a power failure. The chance of getting a corrupted file system is very rare, but exists.

Good to know:

For a system with FBWF disabled, use a NTFS formatted file system.

7 System Errors - Blue Screens

7.1. Overview

If the Windows XP kernel detects an unrecoverable error (e.g. an NMI interrupt is pending), then the kernel issues a “blue screen” to inform the user of the faulty hardware or malfunctioning software. In most cases there are badly written drivers involved, because drivers can run in “kernel mode” and thus have privileges and rights to access and damage critical kernel sections.

The “blue screen” not only informs the user of a system crash, but also it has the responsibility to halt the processor. Like this, the whole system is physically stopped and there could be no more damage than already has been done.

7.2. Enable Auto Reboot

An embedded system might call for a different behavior than just halting the processor. That's where the AutoReboot feature comes into play. With AutoReboot enabled, the system issues a reboot every time a “blue screen” is happening. This results in a system that might recover from a “blue screen” without user interaction.

To enable AutoReboot, press *WINKEY+Break*, change to *Advanced Tab* and click the “*Startup and Recovery*” button. In the dialog, check “*Automatically restart*”. After the next reboot, the “AutoReboot” feature is enabled.

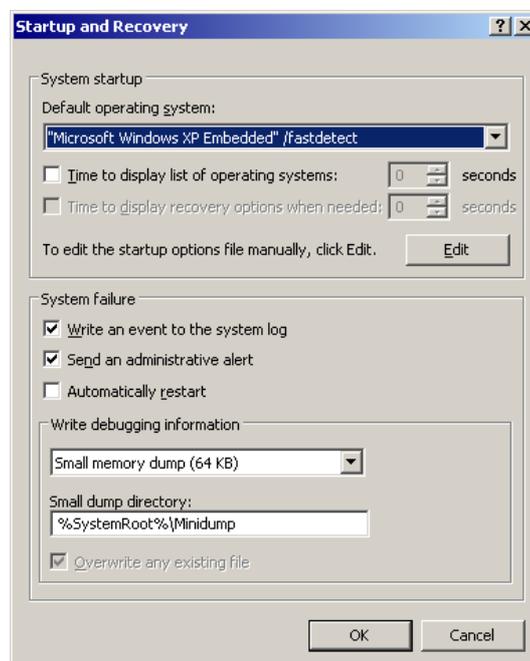


Figure 8 - Auto Reboot feature

8 Remote Debugging

Application debugging can be done from the Visual Studio .NET 200x integrated development environment running on a distant host. The debug session runs over an Ethernet connection. The application and its source files are built, debugged and executed on the target.

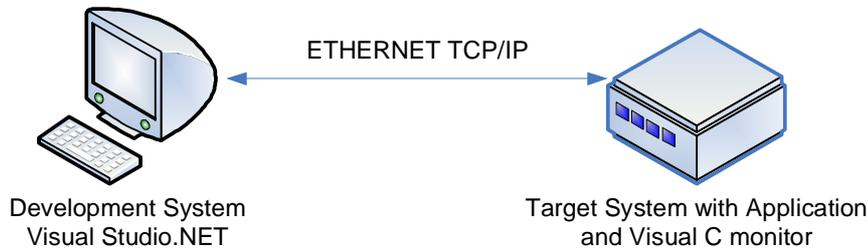


Figure 9 - Remote Application Debugging

A monitor application also runs on the target and communicates symbol information back to the Visual Studio debugger. The XP Embedded target must share the application folder with the Development system (map network drive).

8.1. Connect and Debug

The following instructions assume that you know the Embedded XP network identifier (get it from *Control Panel/System/Computer Name*, e.g. Wes09) of your embedded system and you have a working Ethernet connection between the Embedded target and the development computer. On the development PC, there's a Visual Studio installed.

Boot the target and try to ping it from a development system's command window:

```
C:\>ping Wes09
Pinging XPE [192.168.1.41] with 32 bytes of data:
Reply from 192.168.1.41: bytes=32 time<10ms TTL=128
```

In the same command window, use NET.EXE to map the remote WES2009 drive C: as a local drive W: into the development machine.

```
C:\>net use w: \\Wes09\c$
The password or user name is invalid for \\Wes09\c$.
Type the password for \\Wes09\c$: netipc
The command completed successfully.
```

Create your embedded application "Sample" with Visual Studio on the shared network drive W:. In the project settings under "Debugging", make the following configurations:

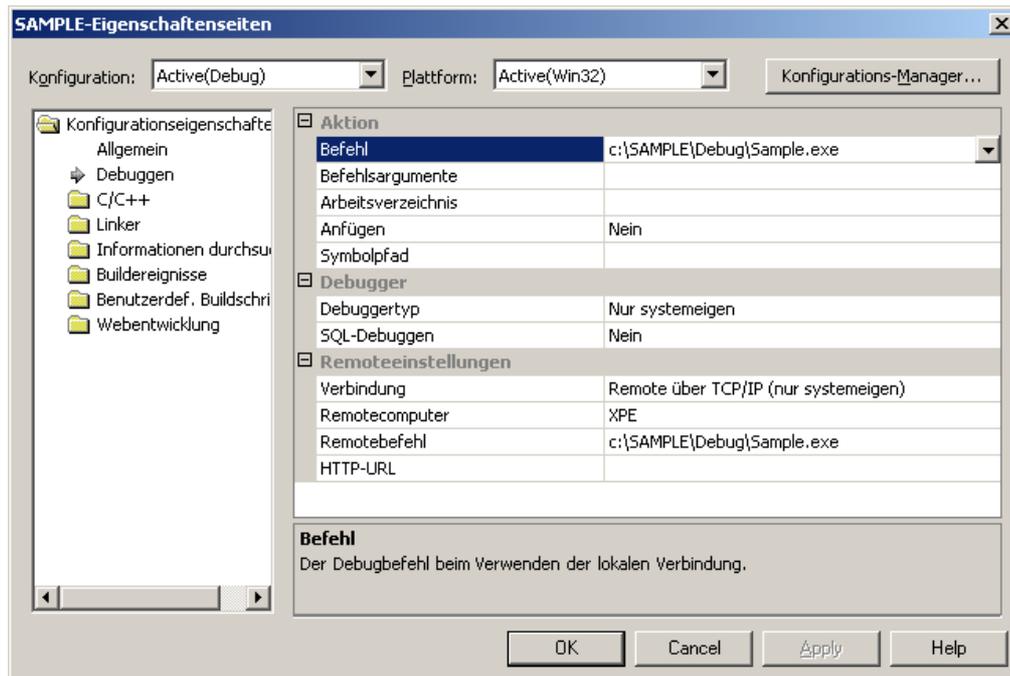


Figure 10 – Remote Debugging Setup

On the embedded WES2009 device, start the debug monitor from within a command shell.

```
C:>cd Documents and Settings\Administrator\Desktop\Utilities\MSVCMON  
C:\ Documents and Settings\Administrator\Desktop\Utilities\MSVCMON>msvcmon -tcpip  
-anyuser.
```

Now you can compile the Project and start the debug session within Visual Studio on your Development system (menu Debug/Start). Enjoy remote application debugging!

9 Installing WES2009

9.1. Overview

If you have lost the Compact flash contents by accident, you must install the WES2009 image again. Please execute the following instructions to re-install WES2009 on the Compact flash card.

9.2. Prepare the CompactFlash for File Copy

The standard WES2009 Image is installed onto a NTFS formatted CompactFlash. The MBR must not be modified in order to boot WES2009. If the CompactFlash is FAT formatted the CompactFlash has to be prepared with the bootprep tool provided by Microsoft.

The easiest way to format the flash device is to use a Windows 7 desktop computer.

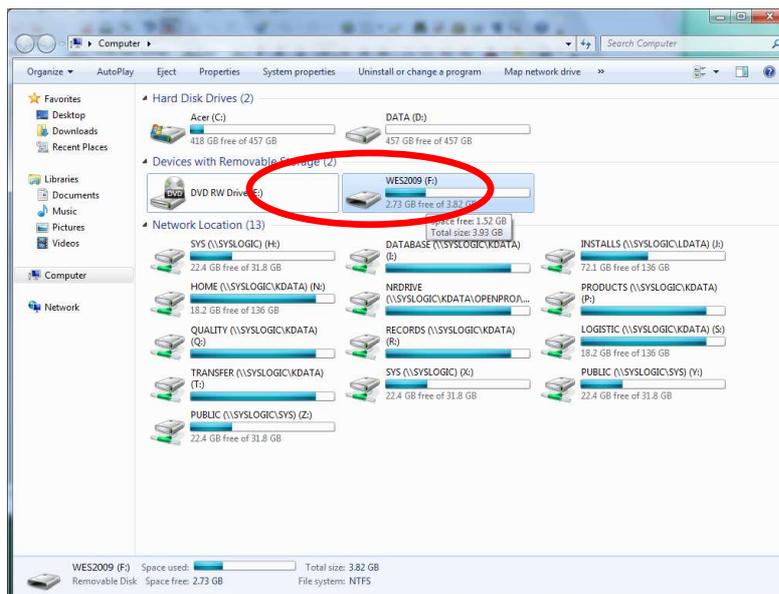


Figure 11 – Windows explorer

Right click the appropriate drive and select *format*. A new windows is shown according to the following figure

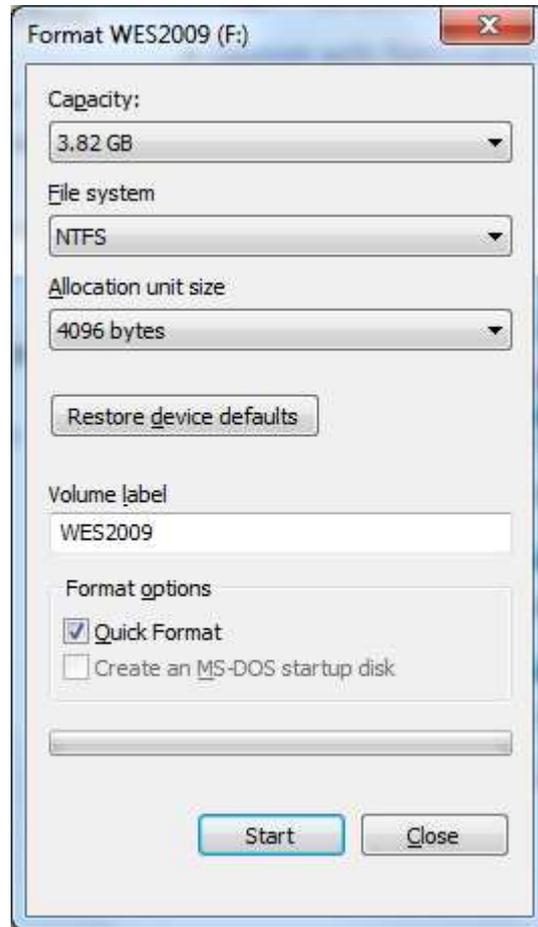


Figure 12 – Format

It's important the file system is set to *NTFS*. After the CompactFlash has been formatted successfully you can now copy the files onto the device.

Note:

Windows only supports to create one partition on a removable device like a CompactFlash Card. To create a CompactFlash Card with additional Data-Partitions you have the following options:

- Use DiskPart to create and format a primary partition with 3GB of size. Copy system files to this partition and boot from it on the target. After booting, create the second Data partition.
- Use PartitonWizard (<http://www.partitionwizard.com/>) to create to partitions on a attached CompactFlash-Card.

9.3. Copy the WES2009 files to the Compact Flash

Now that the Compact Flash is formatted you need to transfer the WES2009 system files. You need a Compact Flash card reader to copy the files from the DVD Drive to the card.

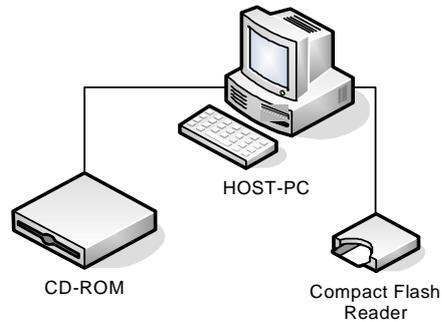


Figure 13 – WES2009 filecopy setup

Copy all the files from the DVD directory `/COMPACTFLASH/FILES_PRE_FBA/` in Windows Explorer to the Compact Flash card after inserting it into the card reader. The copy process may take a while.

Note:

It is important to copy the following two files first:

NTDETECT.COM

ntldr

After that you can copy the rest of the system files and directories to the flash card.

After all files have been copied successfully the CompactFlash has to be removed from the reader. Right click the drive letter and select *eject*. and put back into the target hardware.



Figure 14 – Eject CompactFlash

Stick the flash device into the target hardware and switch on the power supply.

9.4. Post FBA setup

When first booting WES2009 after installing it, there's a setup program called FBA (First Boot Agent) running that takes a lot of time to complete (~60 minutes). Do not interrupt the FBA or the system will get into an unstable state. After the FBA has set up everything WES2009 will boot normally. After that your WES2009 system is ready to go...

10 WinIo Driver Library

10.1. WinIo Library Overview

WinIo is a driver that grants your application access to Windows XP/2000/NT system resources that are normally not accessible. You can use WinIo to simply write a byte to the x86 IO port registers or access physical memory directly. WinIo is written and maintained by Yariv Kaplan (www.internals.com, see 10.5 for license details).

WinIo has the following function interface defined (listing not complete):

```
bool _stdcall InitializeWinIo()
```

This function initializes the WinIo library

void _stdcall ShutdownWinIo()

This function performs cleanup of the WinIo library.

bool _stdcall InstallWinIoDriver(PSTR pszWinIoDriverPath, bool IsDemandLoaded);

This function installs the WinIo driver.

bool _stdcall RemoveWinIoDriver()

This function removes the WinIo driver from the system.

bool _stdcall GetPortVal(WORD wPortAddr, PDWORD pdwPortVal, BYTE bSize);

This function reads a BYTE/WORD/DWORD value from an I/O port.

bool _stdcall SetPortVal(WORD wPortAddr, DWORD dwPortVal, BYTE bSize);

This function writes a BYTE/WORD/DWORD value to an I/O port.

10.2. NETIPC extensions to WinIO

Syslogic has extended the original WinIo Library with some hardware-related functions. These extensions can be accessed like any other WinIo function after properly initializing the WinIo driver:

bool _stdcall GetTemp(float& value)

This function returns the temperature from the on-board LM75 temperature sensor.

bool _stdcall WatchdogGetState(bool& Timeout)

This function returns the state of the Watchdog. 1=Timeout occurred, 0=No Timeout

bool _stdcall WatchdogEnable()

This function enables the watchdog. It will reset the board if not triggered by WatchdogTrigger.

bool _stdcall WatchdogTrigger()

This function triggers the Watchdog.

bool _stdcall WatchdogHardReset()

This function resets the watchdog by hard resetting the board.

bool _stdcall GetCpuFID(DWORD& FID)

bool _stdcall GetCpuOID(DWORD& OID)

bool _stdcall GetCpuRID(DWORD& RID)

These functions return the board identifiers.

10.3. Using WinIo in a Visual C++ application

In order to use WinIo in a Visual C++ application, you should take the following steps (please also have a look at the the TestProg Sample Application found in the TestProg directory):

- (1) Place winio.dll, winio.vxd and winio.sys in the directory where your application's executable file resides.
- (2) Add winio.lib to your project file by right clicking on the project name in the Visual C++ workview pane and selecting "Add Files to Project...".
- (3) Add the #include "winio.h" statement to your source file.
- (4) Call InitializeWinIo.
- (5) Call the library's functions to access I/O ports and physical memory.
- (6) Call ShutdownWinIo.

10.4. TestProg Sample Application

The TestProg sample application demonstrates the use of the extended WinIo Library. After reading out the temperature and the board identifiers, it activates the watchdog and triggers it for about 10 seconds before letting the watchdog generate a timeout.

Copy the WINIO/Bin directory from the DVD-ROM to the Windows WES2009 device and run TestProg.exe (WinIo.dll and WinIo.sys must be in the same directory as TestProg.exe).

10.5. WinIO End User License Agreement

Software License Agreement for WinIo

The following terms apply to all files associated with the software unless explicitly disclaimed in individual files.

IMPORTANT- PLEASE READ CAREFULLY: BY INSTALLING THE SOFTWARE (AS DEFINED BELOW), OR COPYING THE SOFTWARE, YOU (EITHER ON BEHALF OF YOURSELF AS AN INDIVIDUAL OR ON BEHALF OF AN ENTITY AS ITS AUTHORIZED REPRESENTATIVE) AGREE TO ALL OF THE TERMS OF THIS END USER LICENSE AGREEMENT ("AGREEMENT") REGARDING YOUR USE OF THE SOFTWARE. IF YOU DO NOT AGREE WITH ALL OF THE TERMS OF THIS AGREEMENT, DO NOT INSTALL, COPY OR OTHERWISE USE THE SOFTWARE.

1. GRANT OF LICENSE: Subject to the terms below, Yariv Kaplan ("AUTHOR") hereby grants you a non-exclusive, non-transferable, non-assignable license to install and to use the downloadable version of WinIo ("SOFTWARE").

a. Redistributable Code. You may reproduce and distribute the object code form of the SOFTWARE solely in conjunction with, and as part of, your application ("Permitted Application"); provided that you comply with the following:

If you redistribute any portion of the Redistributable Code, you agree that: (i) you will only distribute the Redistributable Code in conjunction with, and as part of, your Permitted Application which adds significant functionality to the Redistributable Code and that distribution of the Permitted Application does not compete with the AUTHOR's distribution of the SOFTWARE; (ii) you will include a valid copyright notice on your Permitted Application; (iii) you will not permit further redistribution of the Redistributable Code; and (iv) you will indemnify, hold harmless, and defend the AUTHOR from and against any claims or lawsuits, including attorneys' fees, that arise or result from the use or distribution of your Permitted Application.

b. License to use Source Code. You may not sell, lease, rent, transfer or sublicense the source code of this SOFTWARE.

2. **MODIFICATION:** SOFTWARE Source Code may be modified without the prior written permission of the AUTHOR. Any modifications made to the SOFTWARE will continue to be subject to the terms and conditions of this AGREEMENT.

3. **COPYRIGHT:** All rights, title, and copyrights in and to the SOFTWARE and any copies of the SOFTWARE are owned by the AUTHOR. The SOFTWARE is protected by copyright laws and international treaty provisions. Therefore, you must treat the SOFTWARE like any other copyrighted material.

4. **TITLE:** You acknowledge that no title to the intellectual property in the SOFTWARE is transferred to you. Title, ownership, rights, and intellectual property rights in and to the SOFTWARE shall remain the exclusive property of the AUTHOR. The SOFTWARE is protected by copyright laws of the United States and international treaties.

5. **LIMITATION OF LIABILITY:** You must assume the entire risk of using the SOFTWARE.

IN NO EVENT SHALL THE AUTHOR BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE, ITS DOCUMENTATION, OR ANY DERIVATIVES THEREOF, EVEN IF THE AUTHOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE AUTHOR SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. THIS SOFTWARE IS PROVIDED ON AN "AS IS" BASIS, AND THE AUTHOR HAS NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

11 Errata

11.1. USB Storage Device Removal

11.1.1. Description

Due to a bug in the USB drivers removable storage devices are treated as fixed drives.

11.1.2. Workaround

1. Rename `c:\windows\system32\drivers\usbhub.sys` → `usbhub_fixed.sys`
2. Rename `c:\windows\system32\drivers\usbstor.sys` → `usbstor_fixed.sys`
3. Copy `usbhub` from a standard Windows XP SP3 installation into the folder mentioned above.
4. Copy `usbstor` from a Standard Windows XP SP3 installation into the folder mentioned above.
5. Restart the device.

12 References

- Microsoft Developer Network
<http://msdn.microsoft.com/en-us/library/ff795587%28v=WinEmbedded.0%29.aspx>
- IPC/SL71 IPC user documentation
(Document Order code: DOC/COMPACT71-SL), available at www.syslogic.com
- IPC/ML71 IPC user documentation
(Document Order code: DOC/COMPACT71-ML), available at www.syslogic.com
- IPC/COMPACT7-SL IPC user documentation
(Document Order code: DOC/COMPACT7-SL), available at www.syslogic.com
- IPC/COMPACT7-ML IPC user documentation
(Document Order code: DOC/COMPACT7-ML), available at www.syslogic.com
- Windows XP Embedded Advanced, by Sean D. Liming
RTC Books, San Clemente CA, U.S.A.
- Microsoft Windows Sysinternals Tools & Website
<http://www.microsoft.com/technet/sysinternals/default.msp>
- Intel Embedded Graphics Driver for Embedded Intel Architecture
<http://edc.intel.com/Software/Downloads/IEGD/>
- Intel Embedded Media and Graphics Driver for Embedded Intel Architecture
<http://edc.intel.com/Software/Downloads/EMGD/>

13 Contact

Our distributors and system integrators will gladly give you any information about our products and their use. If you want to contact the manufacturer directly, please e-mail a message containing a short description of your application and your request or use one of our request forms on our homepage.

Syslogic Datentechnik AG
Switzerland

E-Mail: information	info@syslogic.com
technical support	support@syslogic.com

Homepage:	www.syslogic.com
-----------	--