Microsoft Systems Management Server 2003 Advanced Client
Support for
Microsoft Windows XP Embedded

**Published**:    October 2007

**Revised**:    March 2008

**Applies To:**    Microsoft Systems Management Server 2003 SP3

**Document Version:**    1.2

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This guide is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft®, Active Directory®, Visual C++®, and Windows® are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2007 Microsoft Corporation. All rights reserved.

Microsoft Corporation • One Microsoft Way • Redmond, WA 98052-6399 • USA

**Table of Contents**

[Introduction 3](#_Toc180484470)

[Using SMS 2003 with Windows XP Embedded 4](#_Toc180484471)

[General Software Update Processes with EWF or FBWF 6](#_Toc180484472)

[Using Windows XPe Write Filters with SMS 2003 7](#_Toc180484473)

[SMS 2003 Advanced Client with EWF 9](#_Toc180484474)

[SMS 2003 Advanced Client with FBWF 10](#_Toc180484475)

[SMS 2003 Advanced Client with Registry Filter 11](#_Toc180484476)

[Other Considerations with Windows XPe 11](#_Toc180484477)

[Known Limitations of SMS 2003 Clients Running on Windows XPe Devices 11](#_Toc180484478)

[Appendix A: Write Filter Script 13](#_Toc180484479)

# Introduction

Microsoft Systems Management Server 2003 (SMS 2003) allows large numbers of Microsoft Windows-based clients to be managed across an enterprise. Beginning with SMS 2003 sp1, the rich management features available in SMS 2003 have been extended to support Microsoft Windows XP Embedded (Windows XPe) devices. Typically, these devices are employed as point of sale (POS) devices, automated teller machines (ATMs), industrial controllers, gateways, Internet Protocol (IP) telephones, server appliances, Windows-based thin clients, advanced consumer electronics, and specialized handheld devices.

The following list outlines the high-level functionality and capabilities supported by the SMS 2003 SP3 Advanced Client running on Windows XPe devices on an SMS 2003 site:

* Hardware Inventory
* Software Inventory\*
* Software Metering
* Software Distribution
* Inventory Tool for Microsoft Updates scanning
* Inventory Tool for Microsoft Updates patching\*
* Remote Tools\*
* Remote Desktop
* Remote Assistance\*
* Reports for the above feature sets

\*For further information, refer to the section, “[Known Limitations of SMS 2003 Clients Running on Windows XPe Devices](#_Known_Limitations)” on page .

**NOTE:** Windows XPe thin clients can use the Enhanced Write Filter or File-based Write Filter, which requires additional steps to persist data and updates. For more information, see “[Using Windows XPe Write Filters with SMS 2003](#_Using_Write_Filters)” on page 7.

# Using SMS 2003 with Windows XP Embedded

### SMS 2003 Advanced Client Prerequisites

To operate successfully, the SMS 2003 Advanced Client requires that the following components are installed as part of the Windows XPe image. If you create your own custom image using Windows Embedded Studio Target Designer, ensure that the required components are included in the image. If you obtain Windows XPe as a static image from a supplier, ensure that the supplier is aware of the components that must be included in the image.

* Active Directory Service Interface (ADSI) Core
* Active Directory Service Interface (ADSI) LDAP Provider
* Active Template Library
* Background Intelligent Transfer Service
* Client/Server Runtime (Console)
* Cluster management Support
* COM+ Services
* Common Control Libraries Version 6 [1.0.0.0]
* Common Control Libraries Version 6 [6.0.0.0]
* Credential Management Interface
* Cryptographic Service Providers
* DHCP Client Service
* Distributed Transaction Coordinator (MSDTC)
* Group Policy Core Administration Snap-In
* Local Security Authority Subsystem (LSASS)
* Microsoft Visual C++ Run Time
* Msxml 3.1
* Primitive: AuthZ
* Primitive: Cfgmgr32
* Primitive: Clbcatq
* Primitive: Comres
* Primitive: Crypt32
* Primitive: Dnsapi
* Primitive: Iphlpapi
* Primitive: Mlang
* Primitive: Mpr
* Primitive: Mprapi
* Primitive: Msasn1
* Primitive: Msi
* Primitive: Mswsock
* Primitive: Ncobjapi
* Primitive: Netapi32
* Primitive: Netman
* Primitive: Ntdll
* Primitive: Ole32
* Primitive: Oleaut32
* Primitive: Psapi
* Primitive: Rasapi32
* Primitive: Secure32
* Primitive: Security
* Primitive: Setupapi
* Primitive: Shell32
* Primitive: Shlwapi
* Primitive: Sxs
* Primitive: Userenv
* Primitive: Version
* Primitive: Winmm
* Primitive: Winrnr
* Primitive: Winspool
* Primitive: Winsta
* Primitive: Wldap32
* Primitive: Wmi
* Primitive: Ws2\_32
* Primitive: Ws2Help
* Primitive: Wsock32
* Primitive: Wtsapi32
* Primitive: Wzsvc
* Remote Assistance Automation Framework
* Remote Assistance Channel
* RPC Local Support
* Security Accounts Manager Client
* Standard Template Libraries (STL)
* System Event Notification Services (SENS)
* TCP/IP Networking
* Urlmon Library
* Windows API – Advanced
* Windows API – GDI
* Windows API – Kernel
* Windows API – User
* Windows Installer Service
* Windows NT Marta Core
* Windows Update Agent
* WinHTTP
* WMI Core
* WMI Tools
* WMI View provider
* WMI Win32 Provider

###

### Understanding Write Filters in Windows XP Embedded

For security and operational reasons, it might be undesirable to write to storage media in Windows XPe devices. Write filters provide the ability to write-protect a run-time image. By redirecting all write requests to either a separate disk partition or RAM, a write filter allows the run-time image to maintain the appearance of a writeable run-time image without actually committing any changes to the storage media.

Starting with Feature Pack 2007 for Windows XP Embedded Service Pack 2, two write filters are available:

* File-Based Write Filter (FBWF), which operates at the file level.
* Enhanced Write Filter (EWF), which operates at the sector level.

EWF and FBWF redirect all writes targeted for protected volumes to a RAM or disk cache called an overlay. Used in this context, an overlay is similar to a transparency overlay on an overhead projector. Any change made to the overlay affects the picture as seen in the aggregate, but if the overlay is removed (when the device is restarted), the underlying picture remains unchanged.

EWF works at the sector level on protected disks and allows you to commit changes so that they persist when the device is restarted. EWF is useful for thin clients that do not need to store cached information or receive frequent updates. Changes made to a system protected by EWF are stored in one or more layers that represent snapshots in time. Applying changes to an image commits all changes made to the system during a specific period of time.

FBWF works at the file level instead of the sector level on protected disks. By default, FBWF will protect the whole disk, but Selective Write Through exceptions can be granted to specific files and folders. Writes to folders which have been granted exceptions will be persisted when the device is restarted.

Windows XPe also features a registry filter, which monitors all writes to the registry and stores changes in a separate layer that is discarded when the device is restarted. For registry writes, the registry filter supports persisting specific registry changes when the device is restarted by using registry monitors. This feature is independent from EWF and FBWF and can be used regardless of your specific configuration.

For more information on Windows XPe SP2 Enhanced Write Filter (EWF), see the web page <http://go.microsoft.com/fwlink/?LinkId=79553>.

For more information on Windows XPe SP2 File-Based Write Filter (FBWF), see the web page <http://go.microsoft.com/fwlink/?LinkId=79554>.

## General Software Update Processes with EWF or FBWF

When you use EWF or FBWF protection on the storage device for a Windows XPe device, all uncommitted writes to the device are lost when the device is restarted. This includes any software or software updates that have been installed. In order to persist changes when the device is restarted, the EWF or FBWF must be turned off prior to performing the change actions or the changes must otherwise be explicitly committed to the drive using the **commit** command. This is required regardless of the methodology used to install software or software updates.

The options for persisting changes when the device is restarted are as follows:

###### Method 1: Disable and then Enable the Write Filter

1. Disable EWF or FBWF**.** Disabling the filter will take effect when the device is next restarted. All changes to the system made prior to the device restart are discarded before EWF is disabled.

**Note:** An alternative to the above step is to use the command **commitanddisable** for EWF which will cause all pending write operations to be committed to the storage device before the filter is disabled. This is not recommended as it will commit an unknown number of changes to the storage device.

1. Restart the device. After the device restarts, all new changes are written directly to the storage device. The storage device remains unprotected until EWF or FBWF is enabled again.
2. The required changes to the system can now be made, such as installing or updating software, and changing system settings.
3. Once the necessary changes to the system have been made, EWF or FBWF can be re-enabled. The filter will be enabled after the next device restart. All changes made prior to the restart are saved on the storage device.
4. After restarting the device, the system returns to a protected state.

**Note:** If this process fails at any point after disabling the write filter, the system is left in an unprotected state.

###### Method 2: Use the Commit Command to Write Changes

1. Restart the device to clear any changes that were made to the protected volume.
2. Make the required changes to the system, such as installing or updating software, and changing system settings.
3. Issue the **commit** command to cause all changes made to the system to be committed to the storage device after the device is next restarted.
4. After restarting the device, any changes to the system are committed, and the system returns in a protected state.
5. If any items that have been installed perform additional write actions after restart, you will need to repeat steps 3 and 4.

Although Method 2 does not completely disable the write filter, both methods will write the entire contents of the overlay to disk, regardless of the source of the changes. As noted above, Method 2 may also require an additional device restart and commit action to account for installations that have post-restart actions such as file replacement or drivers and services being installed. With Method 1, you can handle any additional restarts as part of your software installation before re-enabling the write filter.

The use of write filters on Windows XPe imposes unique requirements on system administrators who are tasked with change management. One option for administrators is to use a script to automate the additional actions needed to complete a software or software updates installation. A sample script for performing these write filter modifications in an automated manner is provided in Appendix A. This script utilizes method 1 as described above and has been fully tested for use with SMS 2003.

# Using Windows XPe Write Filters with SMS 2003

When Windows XP Embedded write filters are not enabled, the SMS 2003 Advanced Client behaves normally[[1]](#footnote-2).

However, when write filters are enabled, there are differences in functionality depending on which write filters are used.

To understand these differences, we first need to look again at how the write filters work, and then understand how the SMS 2003 client and its dependent services store information on the Windows XP Embedded device.

### SMS 2003 Advanced Client Components

The SMS Advanced Client is composed of more than a single service executable waiting for instructions from some outside management server. In order to provide a full set of features, as well as scalability and consistency across many platforms, the SMS 2003 Advanced Client utilizes the services of Windows XP features such as WMI, Background Intelligent Transfer Service (BITS), and Windows Server Update Services (WSUS). SMS, along with these other components, store installation and state information in the registry as well as on the primary storage device (disk, RAM disk).

In addition, other directories, files and registry keys may be modified during the installation of software or software updates on a client.

The write filters are designed to take registry and file changes and store them in an overlay, and then discard that overlay when the client is restarted. This poses a challenge to the operation of the SMS Advanced Client and similar products. Depending on which write filter you have employed on your thin client, the implications are different for the SMS Advanced Client and applications in general.

##  SMS 2003 Advanced Client with EWF

The SMS 2003 client makes frequent changes to the registry and to the drive as part of its normal activities, including hardware and software inventory, policy download and evaluation, software distribution and software updates management. Since EWF works at the sector level on protected disks, SMS would be unable to guarantee that only approved changes are committed to the XPe system. For this reason, SMS will not automatically manipulate your write filter configuration in order to persist its own write operations. Instead, SMS administrators will need to employ one of the two methods described earlier in this document as part of their software distribution and software updates management deployments. The next sections describe in more detail specific limitations on SMS when EWF is used on a thin client device.

#### Software delivery and software updates

As mentioned, the write filter must either be disabled before installing software and software updates, or the writes must be explicitly committed to the storage device via the “commit” command. This can be accomplished through use of the enclosed script as described later in Appendix A.

#### Ongoing Client Operation

As a part of its normal operation, the SMS 2003 Advanced Client will connect with the SMS server and download policy information, download and cache files for software installation, collect and send inventory, and send status messages. In addition, it relies on external processes such as WMI and BITS, which also store information on the drive. Since the write filter provides a transparent shield on the system, these processes continue to happen normally to the system. However, at the next restart, unless data is committed to the drive, all of these changes are lost. This means that when the client restarts and loads Windows, the SMS 2003 Advanced Client will begin to operate as if it hasn’t performed any of those operations since the last time the state data for those operations were committed to disk.

As a result, the following are likely to occur:

* The Windows XPe client will re-download all SMS policies which were rolled back as a result of the restart.
* The Windows XPe client will re-download from the SMS Distribution Point any software distribution and software updates content that were removed as a result of the restart.
* The Windows XPe client will produce new delta Software and Hardware Inventory reports. Depending on the currency of the data found during those scans, the server may additionally request a full scan to resynchronize the client and server.
* The Windows XPe client will re-send status messages regarding the above activities to the SMS server.

Depending on your specific configuration, you may experience a temporary impact to client performance during initial startup on Windows XPe systems. You may also experience an increase in network traffic to and from the Windows XPe client until the client is again re-synchronized and at a steady operational state.

**Note:** To avoid unnecessary spikes of network traffic, it is recommended that thin clients be restarted only as needed and otherwise left in a powered on state.

## SMS 2003 Advanced Client with FBWF

Like EWF, FBWF protects the drive against changes. FBWF works at the file level instead of the sector level on protected disks. By default with FBWF the whole disk is protected, but certain folders can be given exceptions, known as Selective Write Through, which allow writes to be persisted across restarts.

#### Software delivery and software updates

Although Selective Write Through can be employed to allow write operations to specific locations which bypass the write protection, exclusions for some folders, such as the Windows system folders, or large parent folders, such as the Program Files folder, are not recommended. Additionally, the folders to be granted exceptions must be known in advance and a restart is required to activate these exclusions. As a result, Selective Write Through can be employed for specific scenarios, but may not be practical as a general method for installing software and software updates. Like EWF, the recommended method is to disable the write filter, make the desired changes, and enable the write filter again. The sample script in Appendix A performs this action for FBWF-protected volumes in addition to EWF protected volumes.

#### Ongoing Client Operation

Selective Write-Through can theoretically be used to persist state data from the ongoing operations of the SMS Advanced Client, WMI and BITS. However, since there is no way to specifically limit the necessary exceptions to just the data needed for SMS, this is not recommended. Therefore, the same conditions apply to FBWF protected Windows XPe systems as described above for EWF protected systems.

## SMS 2003 Advanced Client with Registry Filter

In addition to the two disk write filters, there is a registry filter available for XPe that protects the registry against changes, discarding uncommitted changes at each reboot. The Registry Filter does not have an enable, disable or commit functions like EWF or FBWF, but it does have a way to persist changes across restarts through the use of monitors. A registry monitor essentially takes changes to the registry and writes them to a temporary file, which is then re-imported to the registry following each restart. As with exceptions to FBWF, the registry exceptions must be known in advance and must be configured while the active write filter, either EWF or FBWF, is disabled.

Please consult with your hardware vendor to see if registry filter exceptions are an available option for your device.

## Other Considerations with Windows XPe

Some vendors provide a RAM Disk for temporary file storage rather than writing directly to the drive. While not important on a disk-based system, it becomes more important on a flash-based system. Non-volatile RAM storage is fast, but provides a limited number of writes to the media. Excessive writes to the media shortens its life span and may eventually lead to failure. Therefore, standard RAM is used as a way to provide temporary storage that does not impact the non-volatile RAM.

The challenge imposed in this scenario is that the RAM Disk may be allocated in such a way that provides insufficient space for the temporary location of installation files. Software installations often require large amounts of temporary storage to uncompress files and to install to their permanent location. If the space allocated for this temporary storage is too small, the installation will fail.

When configuring automated installation of software, always keep in mind the available temporary storage area and the requirements of the program for temporary storage. If the requirements exceed the allocated space, your installation will need to temporarily reassign the path of *TEMP* and *TMP* system variables to another location. One option is to redirect to a network share.

**Note**: Before redirecting *TEMP* and *TMP* system variables to a network share, ensure the system will have the right access permissions and consider any network latency issues you may encounter connecting to and from the remote share.

# Known Limitations of SMS 2003 Clients Running on Windows XPe Devices

* Operating System Deployment (OSD) is not supported.
* When defining software inventory rules in the **Inventory Collection** tab of the SMS **Software Inventory Client Agent Properties** dialog box, ensure the option to **exclude encrypted and compressed files** is unchecked.
* Software Distribution programs must be configured to run with the option: **Whether or not a user is logged on**.
* The Inventory Tool for Microsoft Updates (ITMU) software updates scanner does not support the **/nobackup** switch when performing installations. As a result, software update install binaries will persist on the target machine. For this reason, Microsoft recommends using SMS software distribution or the Inventory Tool for Custom Updates (ITCU) for deploying software updates to the devices using the **/nobackup** switch. Alternatively, a script can be deployed via SMS software distribution to periodically remove any remaining software update install binaries.
* Known issues with SMS 2003 Remote Tools:
	+ The File Transfer functionality of remote tools is not supported if the target image is compressed. [Note: It will not display the compressed files & as a result will overwrite an existing file without warning.]
	+ The Chat functionality can initiate a session even when no user is logged on into the target machine. Also, unless the user is logged in to a domain, the Chat screen header bar will display "Not logged in".
	+ The Remote Execute feature cannot process VBscripts.
* Known issues with Remote Assistance:
	+ The Remote Assistance feature will detect **no user log-in** and show the correct username whether it is local or domain.
	+ The Remote Assistance Talking feature is not supported.

# Appendix A: Write Filter Script

This script should be used with SMS 2003 to automate configuration of the Windows XPe write filter before and after the installation of software and updates.

**Note:** The sample script has been written to run safely on any machine, including non-Windows XPe computers, but it has been tested only on Windows XPe images. You should test thoroughly before running the script on other systems.

To use the script and perform software distribution and software updates management using Configuration Manager 2007 on Windows XP Embedded devices that use the EWF or FBWF write filter, you must take the following steps:

1. Create a software distribution package and program to set the default scripting host to cscript.
2. Create a software distribution package and program to disable the write filter. Make this program dependent on the Configure Default Scripting Host program created in step 1.
3. For each software distribution and software updates program intended for Windows XPe computers, make them dependent on the Disable Write Filter program created in step 2.
4. For each software distribution and software updates program intended for Windows XPe computers, create a software distribution package and program to re-enable the write filter. Make this program dependent on your software distribution or software updates program created in step 3.
5. Advertise the new Enable Write Filter program created in step 4.

#### To create a software distribution package and program to configure the default scripting host

By default, the default script engine on Windows XPe clients is configured to allow only one parameter. The following steps will configure clients to support multiple parameters.

1. In the SMS 2003 administrator console, right-click on the **Packages** node, click **All Tasks** and then click **Distribute Software**.
2. On the first page of the **Distribute Software Wizard**, click **Next** to bypass the introduction.
3. On the **Package** page of the **Distribute Software Wizar**d, select **Create a new package and program**, and then click **Next**.
4. On the **Package Identification** page, specify a name and optionally other descriptive details for the package and then click **Next**.
5. On the **Source Files** page, select **This package does not contain any files** and then click **Next**.
6. On the **Program Identification** page, specify a name and optionally a comment for the program to configure the default scripting host. Specify the command line associated with this program as:

wscript.exe //H:cscript

1. Click **Next** to continue.
2. On the **Program Properties** page, from the **Program can run** pull-down menu, select **Whether or not a user is logged on**. From the **After running** pull-down menu, select **No action required**. Click **Next** to continue.
3. On the **Advertise a Program** page, select **No** and then click **Next**.
4. On the **Completing the Distribute Software Wizard** page, review the settings and then click **Finish** to create the package and program.

#### To create a software distribution package and program to disable the XPe write filter.

**Create the package and program**

1. Copy the VBScript below and save it in a text file named **ewf-fbwf.vbs**.
2. In the SMS 2003 administrator console, right-click on the **Packages** node, click **All Tasks** and then click **Distribute Software**.
3. On the first page of the **Distribute Software Wizard**, click **Next** to bypass the introduction.
4. On the **Package** page of the **Distribute Software Wizar**d, select **Create a new package and program**, and then click **Next**.
5. On the **Package Identification** page, specify a name and optionally other descriptive details for the package and then click **Next**.
6. On the **Source Files** page, select **Always obtain files from a source directory** and then click **Next**.
7. On the **Source Directory** page, specify the location where you have stored the VBScript and then click **Next**.
8. On the **Distribution Points** page, select which distribution points you want to copy the script package to. Click **Next** to continue.
9. On the **Program Identification** page, specify a name and optionally a comment for the program to disable the write filter. Specify the command line associated with this program as:

 cscript.exe //B ewf-fbwf.vbs DISABLE

1. Click **Next** to continue.
2. On the **Program Properties** page, from the **Program can run** pull-down menu, select **Whether or not a user is logged on**. From the **After running** pull-down menu, select **SMS restarts the computer**. Click **Next** to continue.
3. On the **Advertise a Program** page, select **No** and then click **Next**.
4. On the **Completing the Distribute Software Wizard** page, review the settings and then click **Finish** to create the package and program.

**Edit the package and program**

1. In the SMS 2003 administrator console, expand the **Packages** node, expand the package you just created to disable the write filter, and then click on the **Programs** node.
2. Right-click on the program and then click **Properties**.
3. On the program **Properties** dialog, click on the **Advanced** tab.
4. On the **Advanced** tab, check the box **Run another program first**. From the **Package** pull-down menu, select the package you created earlier to configure the default scripting host. From the **Program** pull-down menu, select the program you created earlier to configure the default scripting host. Check the box **Run this other program every time** and then click **OK**.

#### Make software distribution and software update programs for XPe dependent on the disable write filter package and program.

This step will be repeated for each and every software distribution and software update program to be installed on Windows XPe computers.

1. In the SMS 2003 administrator console, expand the **Packages** node. Expand the package you wish to advertise to Windows XPe computers and then click on the **Programs** node.
2. Right-click on the program and then click **Properties**.
3. On the program **Properties** dialog, click on the **Advanced** tab.
4. On the **Advanced** tab, check the box **Run another program first**. From the **Package** pull-down menu, select the package you created earlier to disable the write filter. From the **Program** pull-down menu, select the program you created earlier to disable the write filter. Check the box **Run this other program every time** and then click **OK**.

**Note:** Do not directly advertise software distribution and software update programs to Windows XPe clients. Use the package and program created in the next section to advertise to Windows XPe clients.

#### To create a software distribution package and program to enable the XPe write filter.

This step will be repeated for each and every software distribution and software update program to be installed on Windows XPe computers.

**Create the package and program**

1. In the SMS 2003 administrator console, right-click on the **Packages** node, click **All Tasks** and then click **Distribute Software**.
2. On the first page of the **Distribute Software Wizard**, click **Next** to bypass the introduction.
3. On the **Package** page of the **Distribute Software Wizar**d, select **Create a new package and program**, and then click **Next**.
4. On the **Package Identification** page, specify a name and optionally other descriptive details for the package and then click **Next**.
5. On the **Source Files** page, select **Always obtain files from a source directory** and then click **Next**.
6. On the **Source Directory** page, specify the location where you have stored the VBScript and then click **Next**.
7. On the **Distribution Points** page, select which distribution points you want to copy the script package to. Click **Next** to continue.
8. On the **Program Identification** page, specify a name and optionally a comment for the program to enable the write filter. Specify the command line associated with this program as:

 cscript.exe //B ewf-fbwf.vbs ENABLE

1. Click **Next** to continue.
2. On the **Program Properties** page, from the **Program can run** pull-down menu, select **Whether or not a user is logged on**. From the **After running** pull-down menu, select **SMS restarts the computer**. Click **Next** to continue.
3. On the **Advertise a Program** page, select **No** and then click **Next**.
4. On the **Completing the Distribute Software Wizard** page, review the settings and then click **Finish** to create the package and program.

**Edit the package and program**

1. In the SMS 2003 administrator console, expand the **Packages** node, expand the package you just created to enable the write filter, and then click on the **Programs** node.
2. Right-click on the program and then click **Properties**.
3. On the program **Properties** dialog, click on the **Advanced** tab.
4. On the **Advanced** tab, check the box **Run another program first**. From the **Package** pull-down menu, select your package for software distribution or software updates. From the **Program** pull-down menu, select the program you want to run for software distribution or software updates. Check the box **Run this other program every time** and then click **OK**.

#### To create an advertisement for Windows XPe computers.

This step will be repeated for each and every software distribution and software update program to be installed on Windows XPe computers.

1. In the SMS 2003 administrator console, right-click on the **Advertisements** node, click **All Tasks** and then click **Distribute Software**.
2. On the first page of the **Distribute Software Wizard**, click **Next** to bypass the introduction.
3. On the **Package** page of the **Distribute Software Wizar**d, select **Select an existing package**. From the **Packages** list, select the package you previously created to enable the write filter and then click **Next**.
4. On the **Distribution Points** page, select which distribution points you want clients to copy the script package from. Click **Next** to continue.
5. On the **Advertise a Program** page, select **Yes** to advertise a program from the package, and then click **Next**.
6. On the **Select a Program to Advertise** page, select the program you previously created to enable the write filter and then click **Next**.
7. On the **Advertisement Target** page, type-in, or browse to an existing collection of Windows XPe computers to which you want to distribute software or software updates. Click **Next** to continue.
8. On the **Advertisement Name** page, specify a descriptive name and comment for the advertisement.
9. On the **Advertise to Subcollections** page, select whether or not you wish to advertise the program to subcollections of the collection you previously specified and then click **Next**.
10. On the **Advertisment Schedule** page, specify any required scheduling options for this advertisement and then click **Next**.
11. On the **Assign Program** page, select **Yes. Assign the program**. Specify any required assignment details and then click **Next** to continue.
12. On the **Completing the Distribute Software Wizard** page, review the settings and then click **Finish** to create the package and advertisement.

Once the clients receive the advertisement, the SMS 2003 client will run each of the packages and programs previously created in the necessary order to install and persist your software distribution or software updates.

#### Important Notes

1. This script assumes that if either EWF or FBWF is installed on the Windows XPe client, that it has been properly configured to protect a specific drive or drives. This script will not create new protection for an unprotected drive. This script only turns the filter off and on as needed for software installation. For FBWF, it is possible to turn on protection but not actually protect a drive.
2. This script will only enable protection for one drive or ALL drives. If your configuration includes EWF protection for multiple drives but not all drives, you will need to modify the functionality of this script accordingly.

Option Explicit

On Error Resume Next

'#--------------------------------------------------------------------------------------------

'#

'# This VBScript is used to allow software installation and update to a Windows Embedded

'# client that is using EWF or FBWF protection on the drive.

'#

'# Full documentation on the use of this VBScript is included in a separate document.

'#

'#--------------------------------------------------------------------------------------------

Dim gWshShell 'WScript.Shell object

Dim strProtectedDrive : strProtectedDrive = null 'The drive shown by EWF or FBWF as being protected

Dim gSystemDrive 'Contains the System Drive (e.g. C:)

Dim strDisableCommand : strDisableCommand = null 'Either the EWF or FBWF command to disable the write filter

Dim strEnableCommand : strEnableCommand = null 'Either the EWF or FBWF command to enable the write filter

Dim bIsWFEnabled : bIsWFEnabled = false 'Boolean for write filter enabled/disabled.

Dim bIsEWFInstalled : bIsEWFInstalled = false 'Boolean for EWF installed

Dim bIsFBWFInstalled : bIsFBWFInstalled = false 'Boolean for FBWF installed

Dim strCommand : strCommand = null 'The string holding the command-line argument

Set gWshShell = WScript.CreateObject("WScript.Shell") '# Create the Shell Object

CheckWindowsEmbedded '# Check to ensure this script is being run on Windows Embedded

' Check if any command line argument is passed and process

If WScript.Arguments.Count = 1 Then

 strCommand = UCase(WScript.Arguments.Item(0))

 Select Case strCommand

 Case "DISABLE"

 CheckWhichWF

 DisableWF

 Case "ENABLE"

 CheckWhichWF

 EnableWF

 Case Else

 WScript.Quit (0)

 End Select

Else

 WScript.Echo "Invalid Command Line: Missing Action Parameter"

 WScript.Echo "USAGE:"

 WScript.Echo "EWF-FBWF.vbs [Disable|Enable]"

 WScript.Quit (0)

End If

WScript.Quit (0)

'--------------------------------------------------------------------------------------------

' CheckWindowsEmbedded

'

' Checks to ensure this script is being run on Windows Embedded, and if not, exits.

' If there was no error in finding the key, read the string values from the key

' If the "EmbeddedNT" string is found, this is XPe.

' In all other cases, bIsXPe is left as false, which causes an exit

'--------------------------------------------------------------------------------------------

Sub CheckWindowsEmbedded()

 Dim arrValues, bIsXPe

 bIsXPe = FALSE

 arrValues = gWshShell.RegRead(XPE\_REG\_KEY)

 If Err.Number = 0 then

 For Each strValue in arrValues

 if strValue = "EmbeddedNT" then

 bIsXPe = True

 end if

 Next

 End If

 Err.Clear

 if not bIsXPe then

 WScript.Quit (0)

 end if

End Sub

'--------------------------------------------------------------------------------------------

' CheckWhichWF

'

' Determines which write filter (EWF or FBWF) is currently in use on this system and sets

' a variable for use in other sections of this script. Since both EWF and FBWF can be on

' a system (but not both controlling the protection of that system), simply testing for

' the existence of the executable is not sufficient. Without using a separate C++ program

' that accesses the EWF or FBWF API, the only way to test for which one is protecting the

' drive is to use the output of the program at the command line to see if it is active

' and which drive is being protected.

'

'--------------------------------------------------------------------------------------------

Sub CheckWhichWF

 Dim oFSO, strFile, strPath, wsx

 Dim output, strLine

 Dim Status

 Set oFSO = CreateObject("Scripting.FileSystemObject")

 strPath = gWshShell.ExpandEnvironmentStrings("%SystemRoot%")

 If IsEmpty(strPath) Then

 WScript.Quit (1)

 Else

 '============================================

 ' EWF

 '============================================

 'check to see if EWFMGR.EXE is on the system

 strFile = strPath + "\system32\EWFMGR.EXE"

 If oFSO.FileExists(strFile) Then

 'run the EWF Manager to see if the drive is actually protected

 set wsx = gWshShell.Exec("%comspec% /c " & strFile & " -all")

 Do

 Status = wsx.Status

 ' parsing the output of the command to see if the EWF is enabled

 ' and which drive is being protected (if any). Typical output

 ' of this command looks like this:

 '

 ' Protected Volume Configuration

 ' Type RAM

 ' State DISABLED

 ' Boot Command ENABLE

 ' Param1 0

 ' Param2 0

 ' Persistent Data ""

 ' Volume ID A7 FA CA 34 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

 '

 output = split(wsx.Stdout.ReadAll(), vbNewLine)

 For Each strLine in output

 if InStr(strLine, "State") then

 ' if the output from the command generated text like above,

 ' EWF must be installed. Now check if it's enabled or disabled.

 bIsEWFInstalled = True

 if InStr(strLine, "DISABLED") then

 bIsWFEnabled = false

 elseif InStr(strLine, "ENABLED") then

 bIsWfEnabled = true

 end if

 End If

 ' Check which drive is being protected, and get the drive letter

 ' since the drive letter is required in the command line

 if (bIsEWFInstalled) AND (InStr(strLine, "Device Name")) Then

 strProtectedDrive = left(right(strLine, 3), 2)

 'set the default commands to be used elsewhere in this script

 strEnableCommand = strFile & " " & strProtectedDrive & " -enable"

 strDisableCommand = strFile & " " & strProtectedDrive & " -disable"

 End If

 Next

 If Status <> 0 Then Exit Do

 WScript.Sleep 10

 Loop

 End if

 '============================================

 ' FBWF

 '============================================

 'If EWF is enabled, FBWF cannot be enabled, so is EWF enabled, skip FBWF

 If (Not bIsWFEnabled) Then

 Dim bCurrentState : bCurrentState = false

 Dim intLoopNum : intLoopNum = 0

 'check to see if FBWFMGR.EXE is on the system

 strFile = strPath + "\system32\FBWFMGR.EXE"

 If oFSO.FileExists(strFile) Then

 'run the FBWF Manager to see if the drive is actually protected

 set wsx = gWshShell.Exec("%comspec% /c " & strFile & " /displayconfig")

 Do

 Status = wsx.Status

 ' parsing the output of the command to see if the EWF is enabled

 ' and which drive is being protected (if any). Typical output

 ' of this command looks like this:

 '

 ' File-based write filter configuration for the current session:

 ' filter state: disabled.

 '

 ' 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: enabled.

 ' protected volume list:

 ' \Device\HarddiskVolume1

 ' write-through list of each protected volume:

 ' \Device\HarddiskVolume1: (none)

 '

 output = split(wsx.Stdout.ReadAll(), vbNewLine)

 For Each strLine in output

 if InStr(strLine, "current session:") then

 ' if the output from the command generated text like above,

 ' EWF must be installed. Now check if it's enabled or disabled.

 bIsFBWFInstalled = True

 ' gathing data from the current session (not the next session)

 bCurrentState = true

 elseif InStr(strLine, "next session:") then

 bCurrentState = false

 bNextState = true

 end if

 ' if the Current State / filter state = enabled, then FBWF is enabled

 if (bCurrentState) AND (InStr(strLine, "filter state:")) then

 if (InStr(strLine, "enabled")) then

 bIsWFEnabled = true

 else

 bIsWFEnabled = false

 end if

 Exit For

 end if

 Next

 If Status <> 0 Then Exit Do

 WScript.Sleep 10

 Loop

 'set the default commands to be used elsewhere in this script

 if bIsFBWFInstalled then

 strEnableCommand = strFile & " /enable"

 strDisableCommand = strFile & " /disable"

 End if

 End if

 End if

 End If

End Sub

'--------------------------------------------------------------------------------------------

' DisableWF

'

' This sub is used to disable the write filter on the drive (EWF or FBWF) so that updates

' can be written to the storage device.

'

' If the drive is already disabled, this subroutine does nothing just return. Otherwise,

' the function will disable the EWF and reboot the machine.

'

'--------------------------------------------------------------------------------------------

Sub DisableWF

 Dim oExec

 ' if the write filter is enabled, disable it

 if bIsWFEnabled then

 Set oExec = gWshShell.Exec(strDisableCommand)

 ' Wait for the previous command to complete.

 Do While oExec.Status = 0

 WScript.Sleep 100

 Loop

 End If

End Sub

'--------------------------------------------------------------------------------------------

' EnableWF

'

' This sub is used to enable the write filter on the drive (EWF or FBWF)

'

' the function will enable the EWF and reboot the machine in two minutes so that task sequence

' will have time to clean up itself.

'

'--------------------------------------------------------------------------------------------

Sub EnableWF

 Dim oExec

 ' if the write filter is installed, enable it

 if (bIsEWFInstalled OR bIsFBWFInstalled) then

 Set oExec = gWshShell.Exec(strEnableCommand)

 ' Wait for the previous command to complete.

 Do While oExec.Status = 0

 WScript.Sleep 100

 Loop

 End If

End Sub

1. Refer to the section “[Known Limitations of SMS 2003 Clients Running on Windows XPe Devices](#_Known_Limitations)”. [↑](#footnote-ref-2)