

Verified for Windows Server™ 2003 Test Specification

VERSION 1.0
April 23, 2003

Microsoft Corporation

This document is provided for informational purposes only and Microsoft makes no warranties, either express or implied, in this document. Information in this document is subject to change without notice. The entire risk of the use or the results of the use of this document remains with the user.

The names of companies, products, people, characters, and/or data mentioned herein are fictitious and are in no way intended to represent any real individual, company, product, or event, unless otherwise noted. Complying with all applicable copyright laws is the responsibility of the user.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in a 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.

Microsoft, MSDN, Windows, and the Windows logo are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Other product and company names mentioned herein may be the trademarks of their respective owners.

© 1998-2003 Microsoft Corporation. All rights reserved.

Welcome

The Verified for Windows Server 2003 Test Specification was developed by Microsoft to identify applications that run successfully on Windows Server 2003. This specification will help software developers by defining the minimum requirements for applications to operate on Windows Server™ 2003. In contrast with the Application Specification for Microsoft® Windows™ 2003 (Certified Applications), this specification does not include features that make applications more robust and manageable, nor does it include features to make applications easier to use or work with the latest technologies.

Top reasons your customers will benefit from an application that meets this compatibility specification:

- The application will install onto Microsoft Windows Server 2003 by means of a robust installation that helps minimize conflicts among shared components.
- The application will run on Microsoft Windows Server 2003 and is unlikely to cause Microsoft Windows Server 2003 to become unstable.
- Customers will be able to cleanly uninstall the application.
- The application meets basic security requirements for Windows Server 2003.

Checklist for Windows Server 2003 Verification

Windows Fundamentals
<ul style="list-style-type: none"> 1.1 Perform primary functionality and maintain stability 1.2 All kernel-mode drivers installed by the application pass verification testing on Windows Server 2003 1.3 All device or filter drivers included with the application have passed Windows HCT testing 1.4 Perform Windows version checking correctly 1.5 Client components and administrative tools
Install/Uninstall
<ul style="list-style-type: none"> 2.1 Do not attempt to replace files that are protected by Windows File Protection 2.2 Non-proprietary files are not overwritten with older versions 2.3 Do not require a reboot inappropriately 2.4 Properly support “Add/Remove Programs” 2.5 Uninstall correctly
Security
<ul style="list-style-type: none"> 3.1 GINA updates and CSPs 3.2 Run in a highly secure configuration 3.3 No additions or modifications are made to the secure desktop 3.4 Compatible with Virus scanning of I/O write to files 3.5 Services running as LocalSystem do not present a UI
Reliability
<ul style="list-style-type: none"> 4.1 Appropriate Resource Use 4.2 Do not cause services to become unavailable

Please note that the “Verified for Windows Server 2003” test is only open to final-release versions of commercially available products. The commercial availability of products will be verified during the test.

Contents

Welcome.....	iii
Checklist for Windows Server 2003 Verification	iv
Contents	v
Windows Fundamentals	1
Install/Uninstall	4
Security Services	7
Reliability.....	9

Windows Fundamentals

Summary of Windows Fundamental Requirements

Customer Benefits

Customers can be confident that a compliant product will execute on Windows Server 2003 and will not adversely affect the reliability of the operating system.

Requirements

- 1.1 Perform primary functionality and maintain stability
- 1.2 All kernel-mode drivers installed by the application pass verification testing on Windows Server 2003
- 1.3 All device or filter drivers included with the application have passed Windows HCT testing
- 1.4 Perform Windows version checking correctly
- 1.5 Client components and administrative tools

How to Comply with Windows Fundamental Requirements

1.1 Perform primary functionality and maintain stability

The application must perform its primary functions without compromising the stability of the operating system or the application. The application may not crash, or cause the operating system to crash while its primary functionality is exercised. Applications will be tested on a computer with two processors, or a processor supporting “hyper threading”. Both types of systems are recognized by the Operating Systems as “dual-processor” or “multi-processor”.

A crash is any failure within the server application that either causes data loss or forces unscheduled downtime of the server or service. In contrast a crash within a client component or utility component is considered to be an application failure that prevents the user from continuing. A failure within a server component or service will not be considered a crash if it meets both of the following conditions:

- a) does not cause loss of data,
- b) does not force shutdown or unscheduled downtime for any server or service.

A failure within a client component or tool will not be considered a crash if it meets all 3 of the following conditions:

- a) does not cause loss of data,
- b) displays information that would allow a typical user to understand what went wrong and how to avoid the problem in the future
- c) allows the user to continue running the application or close it.

Example: If the application creates, edits, and saves multi-page documents, it must not crash or stop responding, and it must not lose the user's data when he or she creates, edits, saves, or opens documents up to the maximum size that you specify.

Example: If the application displays information about the folders on the user's hard drive, it must not crash or destroy data if the user performs, in any order, any of the menu functions in the application, such as creating new folders, moving them, or renaming them.

Users must be able to use system features supported by Windows Server 2003 with the application.

Example: Windows Server 2003 supports mice with more than three controls (buttons). The application must not crash, stop responding, or lose data when the user presses any button or uses any control, such as a wheel, on a supported mouse.

Example: The application must not expect Windows Server 2003 system files or temporary folders to be on any particular drive letter by default or have any maximum or minimum size. Windows Server 2003 may be installed on drive letters other than C or D. The C or boot drive can be quite small, under 100 megabytes (MB).

Example: File and printer names can be long, and Windows supports Latin and non-Latin in these names. The application must not crash or lose data if a user attempts to use long file or printer names.

Example: If the application uses devices, it must not crash if a device is not installed. For example, if a user tries to print when a Printer, Fax, or other output device is not installed, the application must degrade gracefully.

1.2 All kernel-mode drivers installed by the application pass verification testing on Windows Server 2003

Poorly written kernel-mode drivers have the potential to crash the system. Therefore, it is critical that any application that includes kernel-mode drivers, such as backup, copy protection and compact disc (CD) burning products, be thoroughly tested to minimize this risk. Note that requirement 1.3 identifies additional requirements for specific types of device and filter drivers.

If the application includes any kernel-mode drivers, each of these drivers must pass validation testing under the Windows Driver Verifier Manager tool (Verifier.exe). With Driver Verifier installed on your kernel-mode components, Driver Verifier must not report any stop error messages, or otherwise cause system instability while the system and your application are fully exercised.

1.3 All device or filter drivers included with the application have passed Windows HCT testing

If the product includes drivers for hardware devices or filter drivers for categories accepted by the Windows Hardware Quality Labs (WHQL), these drivers must pass the related tests provided in Windows Hardware Compatibility Test (HCT).

The following types of drivers must pass WHQL testing prior to submission for the "Verified for Windows Server 2003" test:

- Hardware Drivers
- Anti-Virus Drivers
- Storage Adapter Drivers (e.g. SCSI, Fibre channel, RAID Controller, RAID/Multi-path IO)
- Network Adapter Drivers (e.g. Ethernet [all types from 10/100 MHz to 2GB], Token Ring, FDDI)
- Additional types as WHQL develops more tests

Note: Tests for these drivers are arranged through WHQL. Additional information is available at <http://www.microsoft.com/hwdq/hwtest/>.

For certain categories of drivers, Windows Server 2003 will present a warning to end users if they attempt to install a driver that does not have a digital signature from Microsoft. For any drivers that are accepted by the Windows Hardware Quality Labs, the component must be digitally signed by Microsoft.

Note: Additional information is available at <http://www.microsoft.com/hwdq/hwtest/pages/digsigs.asp>.

Proof of WHQL testing for all required device and filter drivers included in the application must be submitted to VeriTest. At no point during testing may any warnings appear about unsigned drivers.

1.4 Perform Windows version checking correctly

The application must verify that the operating system meets the minimum version requirements for the application or display a message to the user when blocking installation or execution that the application is not designed for the later Windows version.

The application must pass the “High Version Lie” test with Microsoft’s Application Verifier tool (AppVerifier).

Note: For more information on AppVerifier, search on “AppVerifier” in the MSDN library: <http://msdn.microsoft.com/library/default.asp>.

1.5 Client components and administrative tools

Server applications can include components that are installed separately, possibly onto separate machines. This could include client components (components that might be executed by non-administrator users on a machine other than the server), and administrative tools. If these components are included in the product being verified, they must also meet the same standard.

Install/Uninstall

Summary of Installation Requirements

Customer Benefits

Customers can be confident that a compliant product will install onto Windows Server 2003 without corrupting any third party applications installed on the system.

Requirements

- 2.1 Do not attempt to replace files that are protected by Windows File Protection
- 2.2 Non-proprietary files are not overwritten with older versions
- 2.3 Do not require a reboot inappropriately
- 2.4 Properly support “Add/Remove Programs”
- 2.5 Uninstall correctly

How to Comply with Install/Uninstall Requirements

2.1 Do not attempt to replace files that are protected by Windows File Protection

The application must not attempt to replace any files that are protected by Windows File Protection (WFP). To ensure that the application does not invoke WFP, it should call **SfcIsFileProtected** when installing any file that it did not create. The Windows Installer service does this automatically.

Protected files include the following files that ship on the Windows Server 2003 product CD:

- Most .SYS, .DLL, .EXE, .OCX, .manifest, and .cat files.
- The following fonts: Micros.ttf, Tahoma.ttf, Tahoma.bd.ttf, Dosapp.fon, Fixedsys.fon, Modern.fon, Script.fon, and Vgaoem.fon.

Note: Some redistributable files, such as specific versions of Microsoft Foundation Classes (MFC) DLLs, are installed by Windows Server 2003 and are protected by WFP.

Protected files form the core of the operating system and it is essential for system stability that the proper versions be maintained. These files can only be updated through service packs, operating system upgrades, Quick Fix Engineering (QFE) hot-fixes, and Windows Update. Applications cannot replace them, and attempting to replace these files by any means other than those listed above will result in the files being restored by the Windows File Protection feature (see the subsection About Windows File Protection, below).

If the application requires newer versions of these components, it must update these components by using a Microsoft Service Pack that installs the required versions or in the case of assemblies install an updated version of the assembly.

Example: When Microsoft publishes an update to DirectX, it will be provided in a package (either a Windows service pack or its own service pack). An application including the updated DirectX must use the package install and not attempt to directly install files from the package. Installing individual files is not allowed. Any attempts to do so would be prevented by Windows File Protection and would result in a poor user experience.

About Windows File Protection

Windows File Protection is a feature of Windows Server 2003 that prevents the unauthorized replacement of essential system files. WFP runs as a background process on Windows Server 2003 and monitors the files listed earlier in this section. When WFP detects that a protected file has been changed, it restores the original.

No Windows File Protection messages may appear at any time during the installation of the application, or while exercising its primary functionality.

Note: Attempting to install components that are under Windows File Protection but have not yet been installed on the system will cause Windows File Protection to install the components. This is correct behavior and not a failure of this requirement.

Note: Application Verifier 2.5 and later makes testing for this requirement much easier and more reliable..

2.2 Non-proprietary files are not overwritten with older versions

On occasion, applications install files that are sourced from third parties. In such instances, it is possible that newer versions of these same files have already been installed on the computer by another application. Therefore, if your application installs non-proprietary files, you must ensure it does not overwrite files already present on the computer with older versions.

The application's installation program must properly check to ensure that the latest file versions are installed. Installing an application must never regress any files that you do not produce or that are shared by applications that you do not produce. Replacing a file with another language version of the same file is equally inappropriate.

2.3 Do not require a reboot inappropriately

In Windows Server 2003, very few installation situations require a reboot. Reboots are unwelcome by customers and, in some situations, can make deploying applications difficult. The application must not require or suggest an unnecessary reboot during or after installation or after application uninstall. Using a deployment package based on Windows Installer (MSI) makes meeting this requirement much easier.

Some situations that require a reboot

- Installing a Windows Service Pack or authorized system redistributable may require a reboot. However, do not assume that a reboot is required after installing a Service Pack.
- Installing a Graphical Identification and Authentication dynamic link library (GINA) requires a reboot. However, applications that install a replacement GINA are not eligible to participate in the "Verified for Windows Server 2003" program, as per [requirement 3.1](#) below.

Situations that do not require a reboot

- Reboot is not required for DLL installation, replacement or registration.
- Updating a service component. If the user or other applications will experience any resource unavailability, you must warn the user that certain services will be stopped while they are updated.
- Replacing an existing file that is in use by an application. You must give the user information about any open applications that have loaded the resource files you are updating so that the user can shut down the application and release those files, allowing file replacement to occur without a reboot. For applications based on the Windows Installer (MSI), this work is all done for you.

If you do require a reboot, and you are certain it is for a valid reason, you must prompt users and allow them the option of deferring the reboot.

If your application requires a reboot, and you have determined it for a valid reason, you must provide technical justification to the test lab for why Windows has to be rebooted in your case. This usually only requires a paragraph or so, but saying “we are installing a driver” or “we are installing a <class> driver” is not acceptable.

2.4 Properly support “Add/Remove Programs”

The user must be able to uninstall applications from the Add or Remove Programs item in the Control Panel. The application must supply your product’s name and the location of your application’s uninstaller, so that the Add or Remove Programs item can obtain information about the application as needed to support uninstall. You can write this information directly to the registry during install or, if you are using an installation system based on the Windows Installer service, you can set these values by using properties in the Windows Installer-based package.

When does this apply?

This requirement is not applicable for an application that does not install — that is, if it executes without installing any components, writing to the registry, modifying the system, or leaving any files on the system other than user created files

2.5 Uninstall correctly

The application’s uninstaller must correctly and fully remove the application.

Except as noted later in this section, the application must remove the following:

- All non-shared application files and folders.
- Shared application files whose reference-count (refcount) reaches zero.
- Registry entries, except for keys that the application created to be specifically shared by other programs and could compromise those applications if removed.
- All shortcuts from the Start menu that the application created at the time of installation.
- The uninstaller itself (unless it is a shared component).

It is acceptable to leave log files that support might need after the fact in order to diagnose issues. User preferences may be considered user data and left behind, but an option to do a completely clean removal or have documentation on a simple method to completely clean up is highly recommended

In general, all user data should be left on the system after removal. If your application removal is about to remove user data, the user must be prompted for confirmation.

Security Services

Summary of Installation Requirements

Customer Benefits

Applications must meet basic security requirements for Windows Server 2003.

Requirements

- 3.1 GINA updates and CSPs
- 3.2 Run in a highly secure configuration
- 3.3 No additions or modifications are made to the secure desktop
- 3.4 Compatible with Virus scanning of I/O write to files
- 3.5 Services running as LocalSystem do not present a UI

How to Comply with Security Services Requirements

3.1 GINA updates and CSPs

An incorrectly written Graphical Identification and Authentication dynamic-link library (GINA) can break Smart Card authentication on the PC.

Cryptographic Service Providers (CSPs) also have the potential to disrupt smart card login on the system.

Testing for these components is beyond the scope of the “Verified for Windows Server 2003” test. Applications that include these components are not eligible to participate in the “Verified for Windows Server 2003” program, and should consider other programs such as “Certified for Windows Server 2003”.

In order to comply with this requirement, installation of the application may not cause any changes or additions to be made to the registry key:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\GINADLL or any sub-key or value.

3.2 Run in a highly secure configuration

Applications must be able to perform all primary functions in a highly secure configuration. A highly secure configuration is a system with the predefined security template HISECWS.INF applied. This security template is included with Windows Server 2003 in the windows\security\templates folder.

To meet this requirement, applications must be able to perform all primary functions on a system with HISECWS.INF applied.

3.3 No additions or modifications are made to the secure desktop

The secure desktop is a tightly controlled User Interface presented in the below situations:

- The change password dialog.
- The locked computer screen.

- The UI presented when no user is logged on.
- The Security dialog on a domain member computer when the CTRL+ALT+DEL keys are pressed.

Most applications make no changes to the secure desktop. The standard UI on the secure desktop can only be modified when adding to or replacing the GINA, but it is possible for services to add dialogs or Windows to the secure desktop.

Because the secure desktop is running in a high access context, testing modifications or additions to the secure desktop is beyond the scope of the “Verified for Windows Server 2003” test. Therefore, applications that make additions or modifications to secure desktop are not eligible to participate in the “Verified for Windows Server 2003” program.

The secure desktop will be visually examined for any additions or modifications to ensure compliance with this requirement.

3.4 Compatible with Virus scanning of I/O write to files

Many device conflicts and application stability issues are exposed when the application runs on a system with kernel mode active virus scanning. Applications must be able to perform all primary functionality while a virus scanner’s kernel mode service is actively scanning all files for viruses at full detection mode. The reverse is also true: the kernel mode active virus scanning service must not be disrupted when applications perform their primary functionality. It is good for applications to also perform correctly while a user mode virus scanning utility is used to scan one or more files, but that is not part of this requirement.

Any application interactions with an active virus scanner service must be appropriate and must be handled gracefully. If active virus scanning has any effect on an application’s functionality, the application must ensure that the benefits of the interaction and the options for the user are both clear.

When does this apply?

This requirement applies as described above for all applications other than virus scanners. A virus scanner must be fully qualified as defined by WHQL standards.

Also, a virus scanner runtime must operate correctly on a system with an active file management service running and with an active firewall.

3.5 Services running as LocalSystem do not present a UI

Services running as LocalSystem must not present a User Interface. When a Window is displayed, the associated message queue allows other processes or users to send it any Windows message which may allow hostile code to be introduced and executed in the process providing the UI. If the process is LocalSystem, the hostile code would execute at LocalSystem instead of the level of the originating user or process.

Application Verifier 2.5 and later makes testing for this easy.

Reliability

Summary of Reliability Requirements

Customer Benefits

Applications must work reliably in the Windows environment.

Requirements

- 4.1 Appropriate Resource Use
- 4.2 Do not cause services to become unavailable

How to Comply with Reliability Requirements

4.1 Appropriate Resource Use

The heap, critical sections, and handles can be misused which results in less reliable applications and failures with subtle circumstances that impact customers but may not be easily reproduced. Each of these items are easily tested to ensure they are not misused.

Your application must be able to perform all of its primary and secondary functionality with no critical errors while running under AppVerifier configured to detect heap corruptions, invalid locks usage (critical section use), and invalid handle usage.

4.2 Do not cause services to become unavailable

Services are software components managed via the Service Control Manager (SCM), and often provide resources to multiple applications and other components.

Your application and application installer must not cause Windows services or services created by others to become unavailable even temporarily (such as a reset). This requirement also applies to any third party service installed or configured by your application or application installer. [Requirement 1.1](#) above already mandates that your application maintain stability of the system, and this is just a further clarification of one aspect of this.

Unless you inform the administrator and await guidance on scheduling the shutdown or reset, the only services your application may shutdown or reset are services that are clearly part of your application and are owned by you.

To ensure compliance with this requirement, the event log will be examined for service shutdown events to determine if any third party services were being stopped that wasn't scheduled by the administrator.