Windows Application Quality Cookbook: A Developer's Guide to Application Compatibility, Reliability, and Performance

**Version 1.1**

**December 1, 2008**

This document familiarizes Application Developers with how to verify the compatibility of their applications with the new operating system and provides an overview of the few known application compatibility issues in Windows 7 and Windows Server 2008 R2. It also points out differences in performance, reliability, and usability, and provides links to detailed white papers and other developer guidance.

**Table of Contents**

Table of Contents

[Introduction 5](#_Toc215075189)

[Compatibility 7](#_Toc215075190)

[General Compatibility 7](#_Toc215075191)

[Operating System Versioning 8](#_Toc215075192)

[Removal of Windows Mail 9](#_Toc215075193)

[Internet Explorer 8 - User Agent String 11](#_Toc215075194)

[Removal of WPDUSB.SYS Driver for Windows Portable Devices 12](#_Toc215075195)

[Microsoft Message Queuing (MSMQ) - SHA-2 Is the Default Hash Algorithm 13](#_Toc215075196)

[Windows Server 2008 R2 Compatibility 14](#_Toc215075197)

[Microsoft Message Queuing (MSMQ) - Removal of Windows 2000 Client Support Service 15](#_Toc215075198)

[New Low-Level Binaries 16](#_Toc215075199)

[64-Bit Only 17](#_Toc215075200)

[WoW64 Is Now an Optional Feature for Server Core 19](#_Toc215075201)

[New Features and Enhancements 20](#_Toc215075202)

[File Library Replaces Document Folder 21](#_Toc215075203)

[User Interface - High DPI Awareness 22](#_Toc215075204)

[Internet Explorer 8 - Data Execution Protection/NX 24](#_Toc215075205)

[User Interface - User Account Control Dialog Updates 26](#_Toc215075206)

[ChooseFont() Win32 Common Dialog 27](#_Toc215075207)

[Compatibility - Application Manifest 29](#_Toc215075208)

[Tools, Best Practices, and Guidance 31](#_Toc215075209)

[Application Verifier 32](#_Toc215075210)

[DISM Replaces pkgmgr, PEImg, and IntlConfg Tools 34](#_Toc215075211)

[Compatibility - Windows Troubleshooting 35](#_Toc215075212)

[Reliability Analysis Component 36](#_Toc215075213)

[Windows Error Reporting Problem Steps Recorder 37](#_Toc215075214)

[Terminal Services 38](#_Toc215075215)

[Network Hang Recovery 40](#_Toc215075216)

[Energy Efficiency Best Practices 41](#_Toc215075217)

[Performance Improvement - Minimizing Unresponsive Services 44](#_Toc215075218)

[Certification for Windows 7 and Windows Server 2008 R2 Applications 45](#_Toc215075219)

# Introduction

Windows 7® and Windows Server 2008 R2® introduce the latest operating system technology and software development platform for use by application developers and enterprises worldwide. As part of further enhancing the security, reliability, performance, and user experience of Windows, Microsoft has introduced many new features, improved existing features, and removed others..

While Windows 7 and Windows Server 2008 R2 are highly compatible with most of their respective applications written for Windows XP®, Windows Server 2003®, Windows Vista®, Windows Server 2008®, Windows Server 2008 R2 and their service packs, some compatibility breaks are inevitable due to innovations, security tightening, and increased reliability. Overall, the compatibility of Windows 7 and Windows Server 2008 R2 with existing applications is high.

This document builds on the concepts embodied in the Windows Vista and Windows Server 2008 Application Compatibility Cookbook (<http://msdn.microsoft.com/en-us/library/bb757005.aspx>). Like it, this document provides you with the means to become familiar with how to verify the compatibility of your applications with the new operating system and provides an overview of the few known application incompatibility issues in Windows 7 and Windows Server 2008 R2. More than that, it also points out differences in performance, reliability, and usability, and provides links to detailed white papers and other developer guidance.

In addition, Microsoft is investing in several new and enhanced features and tools to enable you to build higher quality applications and to troubleshoot when applications do not function properly on Windows 7 and Windows Server 2008 R2. The *New Features and Enhancements* and *Tools, Best Practices, and Guidance* sections of this document provide information about each of these.

We take your suggestions for improvements to this document seriously. Please send your feedback to [ISVALLUP@microsoft.com](mailto:ISVALLUP@microsoft.com).

# Compatibility

This section of the document describes those changes in the operating system that you should pay special attention to due to the potential impacts on existing applications and how new applications should be designed. Compatibility contains two subsections: General Compatibility and Windows Server 2008 R2 Compatibility.

## General Compatibility

General Compatibility contains topics that could have impacts on clients or on both clients and servers. Below is the list of topics covered in this section:

### Operating System Versioning

#### Affected Platforms

**Clients** - Windows 7

**Servers** - Windows Server 2008 R2

#### Feature Impact

**Severity** - High

**Frequency** - High

#### Description

The internal version number for Windows 7 and Windows Server 2008 R2 is 6.1. The GetVersion function will now return this version number to applications when queried. This is especially important for AntiVirus, backup, utility applications, and copy protection.

#### Manifestation of Impact

The manifestation of this change is application-specific:

* Any application that specifically checks for the OS version will get a higher version number
* Application installers might prevent themselves from installing the application, and applications might prevent themselves from starting
* Applications might warn users and continue to function properly
* Some applications might become unstable or crash

#### Mitigation

Most applications will function properly on Windows 7 and Windows Server 2008 R2 because the application compatibility in Windows 7 and Windows Server 2008 R2 is very high. However, Windows 7 and Windows Server 2008 R2 include a Compatibility View for installers and applications that check for OS version.

Users can right-click the shortcut or the executable file and apply the Windows XP SP2 or Windows Vista Compatibility View from the Compatibility tab. In most cases, this should enable the application to operate properly without the need for any changes to the application.

#### Solution

Generally, applications should not perform OS version checks. If a specific feature is needed, it is preferable to attempt to find the feature, and fail only if the required feature is missing. At a minimum, applications should always accept version numbers greater than or equal to the lowest supported version of the OS. Make exceptions only when there is a specific legal, business, or system-component need.

### Removal of Windows Mail

#### Affected Platforms

**Clients** - Windows 7

**Servers** - Windows Server 2008 R2

#### Feature Impact

**Severity** - High

**Frequency** - High

#### Description

Microsoft is deprecating the Windows Mail utility and disabling the API CoStartOutlookExpress. The other mail APIs have been marked as deprecated and are slated for removal in a later Windows version. However, the publicly documented APIs that are not marked as deprecated or obsolete will continue to function in Windows 7. Binaries will remain on the users' systems and will continue to be accessible via the APIs, specifically in the cases mentioned above. In addition, the users' e-mail (.eml) and news (.nws) files will remain on the system.

#### Manifestation of Impact

Removal of Windows Mail results in the following:

* All entry points to Windows Mail and Contacts (for example, Start Menu, user-created Shortcuts, Start -> Run, etc.) are removed or disabled. Some of these are removed completely, others will fail when trying to launch.
* All DLLs ship in the box
* Publicly documented APIs continue to work as they did in Windows Vista
* Any APIs that attempt to launch the main browser UI have been modified to create a silent failure. The function will return success, but will not show the UI to the user. APIs that call other dialog boxes (for example, the Spooler or the Accounts dialog) continue to show that UI
* Protocol (mailto, ldap, news, snews, nntp) handlers will not be associated with Windows Mail or Contacts. When attempting to launch these, customers will see an error dialog pointing them to the location where they can set these associations to another program.
* File associations (.eml, .nws, .contact, .group, .wab, .p7c, .vfc) are broken or disabled. When attempting to open a file with these extensions, customers will get a dialog box offering them other apps that are installed that they can use and point them to a Web page that offers solutions.
* Any user files (for example, contact files or messages) remain on the system in the upgrade scenario
* The Contacts folder is hidden by default so customers will not see it
* APIs are marked as deprecatedin MSDN
* The file preview function is removed
* Shell hooks in the right click menu are removed
* The file search function is removed

#### Mitigation

Users should install Windows Live Mail or any other mail product that is able to read .eml and .nws files.

#### Solution

Detect if there is a default mail handler installed. If not, advise user to install Windows Live Mail or any other product that is able to read .eml and .nws files.

Do not design code that calls the Windows Mail UI API, since it will not work. You must find other ways to access the .eml and .nws files. In addition, as soon as is feasible, discontinue your reliance on all other Windows Mail APIs.

#### Compatibility, Performance, Reliability, and Usability Testing

Exercise your application in a Windows 7 environment to ensure that the application does not try to call the UI API.

### Internet Explorer 8 - User Agent String

#### Affected Platforms

**Clients** - Windows XP | Windows Vista | Windows 7

**Servers** - Windows Server 2003 | Windows Server 2008 | Windows Server 2008 R2

#### Feature Impact

**Severity** - Medium

**Frequency** - High

#### Description

The User Agent String is the Internet Explorer identifier that provides data about its version and other attributes to Web servers. Many Web applications rely on, and piggyback on, the Internet Explorer User Agent String. A change in the User Agent String impacts those applications that do so and that depend on an earlier version number. The User Agent string now includes the string 'Trident/4.0' in order to allow differentiation between the Internet Explorer 7 User Agent String and the Internet Explorer 8 User Agent string when running in Internet Explorer 7 Compatibility View. See [Understanding User Agent Strings](http://msdn.microsoft.com/en-us/library/ms537503(VS.85).aspx) for details.

#### Manifestation of Impact

There are two impacted areas:

1. Web pages that explicitly check the User Agent String and do not support the Internet Explorer 8 User Agent String may not run properly. In the majority of cases, this means the pages will be block users from the content they are attempting to access or display incorrect or malformed content.
2. Applications that host Trident (see [Hosting and Reuse](http://msdn.microsoft.com/en-us/library/aa752038(VS.85).aspx)) will default to Internet Explorer 7 using the Web Optional Component, but will not have access to Internet Explorer 8 features.

#### Solution

Ensure that your applications properly handle the new 'MSIE 8.0' version in the User Agent String.

You may also opt in to the Internet Explorer 7 Compatibility View for those applications based on Internet Explorer 7. This can be done with meta tags. See the discussion in [Understanding User Agent Strings](http://msdn.microsoft.com/en-us/library/ms537503(VS.85).aspx) for details.

#### Compatibility, Performance, Reliability, and Usability Testing

Run your applications and Web pages in an Internet Explorer 8 environment on Windows Vista or Windows XP to ensure that they behave in the desired manner.

#### Links to Other Resources

* **Understanding User Agent Strings**: <http://msdn.microsoft.com/en-us/library/ms537503(VS.85).aspx>
* **The Internet Explorer 8 User-Agent String**: <http://blogs.msdn.com/ie/archive/2008/02/21/the-internet-explorer-8-user-agent-string.aspx>
* **User-Agent String and Version Vector**: <http://code.msdn.microsoft.com/ie8whitepapers/Release/ProjectReleases.aspx?ReleaseId=531>
* **Hosting and Reuse**: <http://msdn.microsoft.com/en-us/library/aa752038(VS.85).aspx>

### Removal of WPDUSB.SYS Driver for Windows Portable Devices

#### Affected Platforms

**Clients** - Windows 7

**Servers** - Windows Server 2008 R2

#### Feature Impact

**Severity** - Low

**Frequency** - Low

#### Description

Microsoft has replaced the kernel mode component of the Windows Vista USB driver stack (WPDUSB.SYS) for Windows Portable Devices (WPD) with the generic WINUSB.SYS driver. Communication with the original WPDUSB.SYS driver was via private I/O Control (IOCTL) codes; we have also removed support of these.

Any consumer of these IOCTL codes would have been responsible for proper interpretation and implementation of the Media Transfer Protocol (MTP). Windows Vista did not support use of these IOCTL codes by third-party applications.

#### Manifestation of Impact

Any application that depended on the availability of these private IOCTL codes would no longer have access to USB-connected MTP devices.

#### Mitigation

Users of an application that depends on the private IOCTL codes must use a different application (or an updated version of the application) to access the USB-connected MTP device.

#### Solution

Applications should use the Windows Portable Devices (WPD) API to find and interact with any WPD Device. Although a significant percentage of WPD devices implement MTP for communication with the PC, WPD is not limited to just MTP devices. In addition, where direct access to the device via the private IOCTLs would have limited the application to communication with only USB-connected devices, use of the WPD API expands the list of connectivity options to other communication protocols (for example, Wi-Fi). In the rare cases when the application must be MTP-aware, the WPD API provides a pass-through mechanism for raw MTP commands.

#### Leveraging Feature Capabilities

The WPD API is supported in Windows XP (via the Windows Format SDK), Windows Vista and Windows 7. The Windows 7 implementation of WPD adds support for MTP over Bluetooth.

#### Links to Other Resources

**Windows Portable Devices**: <http://msdn.microsoft.com/en-us/library/ms740786(VS.85).aspx>

### Microsoft Message Queuing (MSMQ) - SHA-2 Is the Default Hash Algorithm

#### Affected Platforms

**Clients** - Windows XP | Windows Vista | Windows 7

**Servers** - Windows Server 2003 | Windows Server 2008 | Windows Server 2008 R2

#### Feature Impact

**Severity** - Low

**Frequency** - Low

#### Description

In Windows 7, MSMQ uses SHA-2 as the default when signing an outgoing message. Additionally, SHA-2 signatures are required for all incoming messages. You can enable support for a lower encryption algorithm through an administrator-accessible registry key.

#### Manifestation of Impact

* MSMQ in Windows 2003 or below will not accept signed messages originating from MSMQ in Windows 7.
* MSMQ in Windows 7 will not accept signed messages originating from Windows 2008 or below.

#### Mitigation

Users should consider upgrading to Windows 7 to leverage the stronger signing algorithm.

To enable seamless signed message exchange between Windows 7 and any down-level operating system, the Administrator must add appropriate exceptions on the MSMQ machines.

## Windows Server 2008 R2 Compatibility

This section contains topics specific to Windows Server 2008 R2. Below is the list of topics covered in this section:

### Microsoft Message Queuing (MSMQ) - Removal of Windows 2000 Client Support Service

#### Platform

**Servers** - Windows Server 2008 R2

#### Feature Impact

**Severity** - High

**Frequency** - Low

#### Description

The Windows 2000 Client Support Service is an optional component of the Message Queuing Server that you can install on a Windows 2003 or Windows 2008 domain controller machine. This service allows Windows 2000 clients to operate in a domain-integrated mode with any Message Queuing server installed on Windows 2003/2008 machines. MSMQ Clients operating on Windows XP upwards do not need this service.

#### Manifestation of Impact

This change impacts Windows 2000 when interoperating in a Windows 7 domain where all domain controllers are Windows Server 2008 R2. If a customer upgrades to the Windows 7 domain, the existing MSMQ applications on any Windows 2000 machines in the domain will not be able to operate in a domain-integrated mode unless these clients upgrade to a higher Windows version.

#### Mitigation

Users who have Windows 2000 Client machines on a Windows 7 domain can configure a Windows 2003/2008 domain controller in the domain and install the MSMQ Windows 2000 Client Support Service on this domain controller.

#### Leveraging Feature Capabilities

Users who have Windows 2000 Client machines running MSMQ should upgrade to a higher Windows version in order to take advantage of the Active Directory-based implementation of the MSMQ Server.

#### Compatibility, Performance, Reliability, and Usability Testing

Users who have Windows 2000 Client machines running MSMQ on a Windows 7 domain with one or more down-level domain controllers should verify that their applications are functional on this mixed domain.

### New Low-Level Binaries

#### Affected Platforms

**Clients** - Windows 7

**Servers** - Windows Server 2008 R2

#### Feature Impact

**Severity** - Medium

**Frequency** - High

#### Description

To improve internal engineering efficiencies and improve foundations for future work, we have relocated some functionality to new low-level binaries. This refactoring will make it possible for future installs of Windows to provide subsets of functionality to reduce surface area (disk and memory requirements, servicing, and attack surface).

#### Manifestation of Impact

As an example of functionality that we moved to low-level binaries, kernelbase.dll gets functionality from kernel32.dll and advapi32.dll. This means that the existing binary now forwards calls down to the new binary rather than handling them directly; the forwarding can be static (the export table shows the redirection), or runtime (the dll has a stub routine that calls down to the new binary). This will affect low-level applications such as security and backup applications that are dependent upon internal APIs and offsets.

#### Solution

The only impact is to code that makes assumptions when attempting to look at the kernel32.dll or the advapi32.dll export table in memory, such as an anti-virus application might do.

Use published APIs and not the details of their implementation. This is just one example of implementing around a detail of implementation for an API.

### 64-Bit Only

#### Affected Platforms

**Servers** - Windows Server 2008 R2

#### Feature Impact

**Severity** - Low

**Frequency** - High

#### Description

Windows Server 2008 R2 ships with a 64-bit SKU only; no 32-bit SKU is available for the server version of the operating system. However, a 32-bit SKU continues to be available for the Windows 7 client.

#### Manifestation of Impact

This will impact three areas:

* 32-bit drivers
* 32-bit plug-ins
* 16-bit executables

#### Solution for 32-bit Drivers

Recompile 32-bit drivers as signed 64-bit drivers.

#### Solution for 32-bit Plug-ins

WoW64, an x86 emulator, allows 32-bit Windows-based applications to run seamlessly on 64-bit Windows. WoW64 is now an optional feature that you must install if it is necessary to run 32-bit code.

The system isolates 32-bit applications from 64-bit applications, which includes preventing file and registry collisions. It supports console, GUI, and service applications. The system provides interoperability across the 32/64 boundary for scenarios such as cut and paste and COM. However, 32-bit processes cannot load 64-bit DLLs, and 64-bit processes cannot load 32-bit DLLs. We commonly see this in shell plug-ins written for Windows Explorer.

A 32-bit application can detect whether it is running under WOW64 by calling the [IsWow64Process](http://msdn.microsoft.com/en-us/library/ms684139(VS.85).aspx) function. The application can obtain additional information about the processor by using the [GetNativeSystemInfo](http://msdn.microsoft.com/en-us/library/ms724340(VS.85).aspx) function

Note that 64-bit Windows does not support running 16-bit Windows-based applications. The primary reason is that handles have 32 significant bits on 64-bit Windows. Therefore, handles cannot be truncated and passed to 16-bit applications without loss of data. Attempts to launch 16-bit applications fail with the following error: ERROR\_BAD\_EXE\_FORMAT.

#### Solution for 16-bit Executables

64-bit Windows recognizes a limited number of specific 16-bit installer programs and substitutes a ported 32-bit version. The list of substitutions is stored in the registry under the following key:

HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows NT\***CurrentVersion***\NtVdm64

There is built-in support for several installer engines, including InstallShield 5.x installers.

Note that the 64-bit Windows Installer can seamlessly install 32-bit MSI-based applications on 64-bit Windows.

#### Links to Other Resources

* **Running 32-bit Applications**: <http://msdn.microsoft.com/en-us/library/aa384249(VS.85).aspx>
* **Performance and Memory Consumption**: <http://msdn.microsoft.com/en-us/library/aa384219(VS.85).aspx>
* **WoW64 Implementation Details**: <http://msdn.microsoft.com/en-us/library/aa384274(VS.85).aspx>
* **Registry Redirector:** <http://msdn.microsoft.com/en-us/library/aa384232(VS.85).aspx>
* **File System Redirector**: <http://msdn.microsoft.com/en-us/library/aa384187(VS.85).aspx>
* **Memory Management**: <http://msdn.microsoft.com/en-us/library/aa384209(VS.85).aspx>
* **Processor Affinity**: <http://msdn.microsoft.com/en-us/library/aa384228(VS.85).aspx>
* **Interprocess Communication**: <http://msdn.microsoft.com/en-us/library/aa384203(VS.85).aspx>
* **Application Installation**: <http://msdn.microsoft.com/en-us/library/aa384143(VS.85).aspx>
* **Debugging WoW64**: <http://msdn.microsoft.com/en-us/library/aa384163(VS.85).aspx>
* **Is WoW64 Process Running**: <http://msdn.microsoft.com/en-us/library/ms684139(VS.85).aspx>
* **Get Native System Info**: <http://msdn.microsoft.com/en-us/library/ms724340(VS.85).aspx>

### WoW64 Is Now an Optional Feature for Server Core

#### Platform

**Servers** - Windows Server 2008 R2

#### Feature Impact

**Severity** - Low

**Frequency** - Low

#### Description

The Server Core installation option for Windows Server 2008 R2 does not install WoW64 by default. Instead, WoW64 is now an optional feature that you can install if it is necessary to run 32-bit code.

In addition, the Active Directory, Active Directory Lightweight Directory Services, and Web Server roles require the installation of WoW64 in order to run in Windows Server 2008 R2.

#### Manifestation of Impact

Administrators running 32-bit code on Server Core will receive an error message that the application cannot be executed.

If Administrators attempt to run Active Directory, Active Directory Lightweight Directory Services, and Web Server, they will receive an error message.

#### Mitigation

Install WoW64.

#### Solution

The preferred solution is to provide a 64-bit version of the code to enable it to run on Server Core without the need to install WoW64.

At a minimum, provide user documentation noting that to run 32-bit code they must install WoW64.

#### Compatibility, Performance, Reliability, and Usability Testing

Verify that all code used is 64-bit.

#### Links to Other Resources

* **WoW64 Implementation Details**: <http://msdn.microsoft.com/en-us/library/aa384274(VS.85).aspx>
* **Debugging WoW64**: <http://msdn.microsoft.com/en-us/library/aa384163(VS.85).aspx>
* **Server Core**: <http://msdn.microsoft.com/en-us/library/ms723891(VS.85).aspx>

# New Features and Enhancements

Windows 7 and Windows Server 2008 R2 include many enhancements to improve both the users' and the developers' experience. A few of them, however, have impacts that developers need to be aware of regarding their existing and new products. Below is the list of new features and enhancements:

### File Library Replaces Document Folder

#### Affected Platforms

**Clients** - Windows 7

**Servers** - Windows Server 2008 R2

#### Feature Impact

**Severity** - Medium

**Frequency** - High

#### Description

Libraries provide a centralized folder-like experience for file storage, search, and access across multiple locations, both local and remote.

The default locations used by common file dialogs (for example, Open, and Save) have been changed from the Document Folder to the Documents Library. The User Interface is unchanged, but the user will now be able to view, browse, and search the Library using various arrangement views. Files are saved into the Library default save location unless the user changes the default save location or chooses a different folder.

Developers could create their own libraries or add locations to existing libraries using the IShellLibrary interface. Users can find libraries by using the Known Folder system (for example, FOLDEDID\_DocumentsLibrary).

#### Manifestation of Impact

The Library is itself a file, and ***not*** a folder. Therefore, path manipulations could result errors due to the attempt by the application to concatenate files to files.

#### Solution

When using IFileDialog, you must use GetResult method instead of combination of GetFolder and GetFilename as you would in the previous OS versions. Use the Shell APIs where possible to interact with and manipulate items in the Shell Namespace (for example, IShellItem).

#### Leveraging Feature Capabilities

If you want to create your own libraries or add locations to existing libraries, you must use IShellLibrary API. Libraries are themselves Shell Folders so you can enumerate them just like any other Shell Folder.

#### Compatibility, Performance, Reliability, and Usability Testing

Using the common file dialog will ensure that users can save directly to their libraries.

#### Links to Other Resources

See MSDN documentation for IShellLibrary

### User Interface - High DPI Awareness

#### Affected Platforms

**Clients** - Windows XP | Windows Vista | Windows 7

#### Feature Impact

**Severity** - Medium

**Frequency** - Medium

#### Description

The goal is to encourage end users to set their displays to native resolution and use DPI rather than screen resolution to change the size of displayed text and images. Windows 7 can auto-detect and configure a default DPI on clean installs on machines configured by their OEMs using DPI settings. There are tools you can use to help design applications that are high-dpi aware in order to ensure the most readable results.

We have added two new High DPI features to Windows 7:

* Per-user DPI setting (previously per machine)
* Change DPI without rebooting (logoff/logon is still required)

#### Manifestation of Impact

Applications that do not handle the high DPI case are likely to exhibit visual artifacts, including:

* Clipping of UI or text by other UI elements
* Inconsistent font sizes
* Off-screen UIs
* Blurring of text or UI
* Broken drag and drop or other inputs
* Rendering of full screen DX applications partially off screen

#### Solution

To make your applications DPI-aware:

**Step 1**: Do a high-level functional test pass, including install and uninstall at the following settings:

|  |  |
| --- | --- |
| **Setting** | **What to look for** |
| 1024x768 @ 120 DPI (125% scaling)  [**Note**: If your app requires 1024x768, then do this test at 1280x960] | This is an effective resolution of ~800x600, so look for UI clipped off the screen or layout issues. Also look for pixilated bitmaps and icons. |
| 1600x1200 @ 144 DPI (150% scaling) | Blurry UI. Verify that all mouse operations work, especially drag-and-drop operations. Also, verify full-screen modes work properly. |
| 1600x1200 @ 144 DPI with DPI Virtualization disabled. | Often buttons and UI will not scale in relation to larger text and there will be significant text clipping. Look for layout issues in general and pixilated bitmaps and icons. |

**Step 2:** Write down all the issues found, including location, screen resolution, and DPI settings, as well as how the application behaves in the other DPI/Resolution configurations for completeness

**Step 3**: Check each issue against the Common DPI Coding Issues

**Step 4**: Assess the cost of making the application fully DPI aware

**Step 5**: Make a list of the High DPI assets required (for example, buttons, icons)

**Step 6**: Work through and fix the list of DPI issues found in Step 1

**Step 7**: Integrate the new assets from Step 5

**Step 8**: Declare your application DPI Aware

#### Compatibility, Performance, Reliability, and Usability Testing

Re-run the DPI Awareness assessment and verify the issues are fixed.

#### Links to Other Resources

* **High DPI Whitepaper**: <http://go.microsoft.com/fwlink/?LinkID=129586>for the PDF version or<http://go.microsoft.com/fwlink/?LinkID=129588>for the Windows XPS version

### Internet Explorer 8 - Data Execution Protection/NX

#### Platform

**Clients** - Windows XP | Windows Vista | Windows 7

**Servers** - Windows Server 2003 | Windows Server 2008 | Windows Server 2008 R2

Internet Explorer 8 will enable DEP/NX protection when run on an operating system with the latest service pack. Windows XP SP3, Windows Server 2003 SP3, Windows Vista SP1, and Windows Server 2008 all have DEP/NX enabled by default in Internet Explorer 8.

#### Feature Impact

**Severity** - Medium

**Frequency** - Low

Typically, any application that runs in Internet Explorer and is not compatible with DEP/NX will crash on startup and will not function. Internet Explorer may crash on startup if add-ons not compatible with DEP/NX are installed.

#### Description

DEP/NX is a security feature that helps mitigate memory-related vulnerabilities. As of Internet Explorer 8, all Internet Explorer processes enable the DEP/NX feature by default.

#### Manifestation of Impact

The Windows Kernel monitors a program's execution. If the Kernel detects an attempt to run code from a memory page that is not marked executable, the Kernel halts execution of the program, resulting in a "crash." This is a security measure to help ensure that memory-related vulnerabilities (for example, buffer overflows) in the application cannot be exploited in order to execute arbitrary code.

#### End-user Mitigation

* Install a later version of the add-on or framework that is DEP/NX compatible.
* Run IE elevated as Administrator and then disable DEP/NX using the checkbox on the Internet Options / Advanced tab labeled "Enable memory protection to help mitigate online attacks."

#### Developer Solution

* Compile applications using latest versions of frameworks (for example, ATL) that are DEP compatible.

#### Leveraging Feature Capabilities

* Use the /NXCOMPAT linker option to indicate DEP/NX compatibility
* Opt your code into other available defenses like stack defense (/GS), safe exception handling (/SafeSEH), and ASLR (/DynamicBase)

#### Compatibility, Performance, Reliability, and Usability Testing

* Test your code with DEP/NX enabled using latest released Internet Explorer version on Windows Vista SP1 or later.
* Alternatively, test with Internet Explorer 7 on Windows Vista after enabling the DEP/NX option. To enable DEP/NX for Internet Explorer 7: Run Internet Explorer as an administrator, then set the appropriate checkbox in the Tools > Internet Options > Advanced tab

#### Links to Other Resources

* **Internet Explorer 8 Security Part I: DEP/NX Memory Protection**: <http://blogs.msdn.com/ie/archive/2008/04/08/ie8-security-part-I_3A00_-dep-nx-memory-protection.aspx>
* **Data Execution Prevention**: <http://msdn2.microsoft.com/en-us/library/aa366553.aspx>
* **New NX APIs added to Windows Vista SP1, Windows XP SP3 and Windows Server 2008 R2**: <http://blogs.msdn.com/michael_howard/archive/2008/01/29/new-nx-apis-added-to-windows-vista-sp1-windows-xp-sp3-and-windows-server-2008.aspx>

### User Interface - User Account Control Dialog Updates

#### Affected Platforms

**Clients** - Windows 7

**Servers** - Windows Server 2008 R2

#### Feature Impact

**Severity** - Low

**Frequency** - Medium

#### Description

In Windows 7, we have standardized the UAC dialog box choices. Previously, users had to select from multiple options, for example, "Continue," "Cancel," etc. Now all dialog boxes give users a simple "Yes" or "No." The dialog layout now also clearly shows the program name, publisher, and origin.

#### Leveraging Feature Capabilities

The application development requirements in Windows 7 for UAC compatibility are the same as in Windows Vista. Windows Vista-compatible applications will interact with UAC in Windows 7 without any modifications. See the User Account Control topics in the [Windows Vista Application Compatibility Cookbook](http://msdn.microsoft.com/en-us/library/bb757005.aspx) for information about how to make Windows XP applications work correctly on Windows 7.

While the UAC improvements for Windows 7 will impact the user's experience, they will not impact the application interface. However, if there is any help content linked to the UAC dialogs, you may need to update the screenshots.

#### Links to Other Resources

**Windows Vista Application Compatibility Cookbook**: <http://msdn.microsoft.com/en-us/library/bb757005.aspx>

### ChooseFont() Win32 Common Dialog

#### Affected Platforms

**Clients** - Windows 7

**Servers** - Windows Server 2008 R2

#### Feature Impact

**Severity** - Low

**Frequency** - Medium

#### Description

Windows 7 includes several updates to the ChooseFont() Win32 common dialog. These fall into two categories:

* Visual refresh of the dialog
* Support for new show/hide fonts feature

The **dialog refresh** updates the standard template to bring the dialog more in line with other dialog layouts in Windows. It introduces WYSIWYG to the font display lists to help users choose fonts. It also includes a link to the Fonts CPL to provide easy access for users wishing to customize their font lists.

**Show/hide fonts** is a new Windows 7 platform feature whereby fonts not appropriate for the current user's language settings (input methods) are not presented by default in font selection lists. Users may customize the fonts that they wish to appear in the Fonts CPL or may disable this feature.

#### Manifestation of Impact

**Dialog visual refresh**

We have introduced two new templates in Windows 7 (one for applications that load version 6 or later of comctl32.dll and another for applications loading earlier versions).

* For application compatibility reasons, these new templates will only be loaded for applications that do not hook the ChooseFont message queue. Applications that hook the message queue will continue to see the old dialog layout.
* Applications that provide their own templates will continue to be able to use them.

Applications that do not get the new templates will see no dialog layout changes from Windows Vista. They should however still get the new WYSIWYG font preview.

**Show/hide fonts**

For all versions of ChooseFont, the dialog will use the current user's show/hide font settings to determine the font list to display. This will result in the display of fewer font lists in most instances.

#### End-user Mitigation

**Show/Hide Fonts:** To disable font hiding, users should go to the Font Settings page in the Fonts CPL and deselect the "Hide fonts based on language settings" checkbox.

#### Developer Mitigation

* **Visual refresh:** Applications that provide their own templates may want to refresh this to be in line with the appropriate new Windows 7 template. The new templates will be available in the Font.dlg template file at RC.
* **Show/Hide Fonts:** Developers may disable this feature by providing an additional flag (CF\_INACTIVEFONTS) in the flags member of the CHOOSEFONT structure. Setting this flag causes all installed fonts to display in the font list.
* **Show/Hide Fonts:** Applications that provide ChooseFont help content may wish to add content to explain why the font list is reduced and direct users to the Fonts CPL to customize their font lists.

#### Compatibility, Performance, Reliability, and Usability Testing

Developers whose applications hook the ChooseFont message queue to customize the dialog should verify that their applications retain all existing functionality.

Applications that heavily trim the font list using flags should ensure that the font list presented remains acceptable.

### Compatibility - Application Manifest

#### Affected Platforms

**Clients -** Windows 7

**Servers -** Windows Server 2008 R2

#### Feature Impact

**Severity -** Low

**Frequency -** Low

#### Description

Windows 7 introduces a new section in the application manifest called "CompatibilityInfo." This section helps Windows determine the targeted Windows versions of the application and provide legacy behavior to legacy applications, and new behavior to new applications.

This manifest update helps Windows provide a highly compatible environment for applications designed for legacy Windows operating systems. This also helps Windows to ensure that your application continues to work well on newer updates and versions. For Example, in Windows 7, the CompatibilityInfo section will ensure that applications designed for Windows Vista will receive Windows Vista behavior when executing on Windows 7.

#### Manifestation of Change

Applications without a CompatibilityInfo section in their manifest will receive legacy behavior by default. For example, on Windows 7, applications without this section in their manifest will receive Windows Vista behavior. Note that this manifest section will have no effect on Windows XP and Windows Vista.

Here is the current list of Windows components that support this legacy behavior in Windows 7:

* API: GetOverlappedResult
* Desktop Windows Manager: Fail/Lock bit blitting
* RPC exception handling
* API: ReadFileEx
* RPC: Thread pool management

#### Leveraging Feature Capabilities

Update the application manifest with the latest Compatibility information for operating system support. The section below describes the additions to the manifest:

* **Name Space:** Compatibility.v1 (xmlns="urn:schemas-microsoft-com:compatibility.v1">)
* **Section name:** CompatibilityInfo (New section)
* **SupportedOS:** GUID of supported operating system. The GUIDs that map to the supported operating systems are:
  + {66666666-6666-6666-6666-666666666666} for Windows Vista: This is the default value for the switchback context.
  + {77777777-7777-7777-7777-777777777777} for Windows 7: Applications that set this value in the application manifest get the Windows 7 behavior.
  + GUIDs for future Windows versions will be generated and posted appropriately

The XML example of an updated manifest is below:

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<assembly xmlns="urn:schemas-microsoft-com:asm.v1" manifestVersion="1.0">

<compatibility xmlns="urn:schemas-microsoft-com:compatibility.v1">

<application>

<supportedOS Id="{77777777-7777-7777-7777-777777777777}"/>

<supportedOS Id="{66666666-6666-6666-6666-666666666666}"/>

</application>

</compatibility>

</assembly>

#### Compatibility, Performance, Reliability, and Usability Testing

1. Test the application with the new compatibilityInfo section and SupportedOS={7…} to ensure that the application works properly using the latest Windows 7 components
2. Test the application with the new compatibilityInfo section and SupportedOS={6…} to ensure that the application works properly using the legacy behavior for certain Windows 7 components
3. Test the application without the new compatibilityInfo section to ensure that the application works properly using the legacy behavior for certain Windows 7 components

# Tools, Best Practices, and Guidance

This section contains aids for Application Developers who want to either confirm the continued compatibility of their existing applications or ensure optimal quality and compatibility for new applications they are designing. Descriptions of the following tools, best practices, and guidance are included in this section:

### Application Verifier

#### Affected Platforms

**Clients** - Windows XP | Windows Vista | Windows 7

**Servers** - Windows Server 2003 | Windows Server 2008 | Windows Server 2008 R2

#### Description

Promote and enforce Application Verifier as a quality gate for all development. It includes several improvements:

* We have provided additional checks to address issues that the Windows Error Reporting team discovered during thread-pool usage.
* We combined 32- and 64-bit versions of the package to address changes in Windows 7, including the needs for testing 32-bit components under a 64-bit version of Windows, as well as for general simplification.
* We have included additional checks for multi-threaded applications, running 32-bit applications on 64-bit Windows, and 133 bug fixes.

These changes should have no impact on users who do not enable the Thread Checks; those who do should receive additional support in discovery and diagnosis of existing thread-pool usage problems.

While there is a slight performance penalty when using this service, because it is not typically run in retail environments the performance levels should remain acceptable.

#### Usage

To deliver reliable Windows applications:

* First, test applications written in unmanaged code with Application Verifier before releasing it to customers
* Second, monitor application failure reports collected by Windows error reporting
* Finally, follow the steps provided by Application Verifier to resolve errant conditions

Thread-pool checks are enabled by default under the "Basics" check heading. As this is included in the default setting, users need only to run Application Verifier on their code with the default settings to leverage the new checks.

Typically, only debug versions run this feature, so performance is not generally an issue. If performance issues arise from the use of this, or any other Application Verifier check, run one check at a time until you have performed all needed checks.

Nearly 10% of application crashes on Windows systems are due to heap corruption. These crashes are nearly impossible to debug after the fact. The best way to avoid these issues is to test with the Page Heap features found in Application Verifier.

Monitor the reliability status of the applications via the [Winqual](https://winqual.microsoft.com/) Web portal. This portal shows the error reports collected via Windows error reporting, so it is easy to identify the most frequent failures. Learn about this at [Windows Error Reporting: Getting Started](http://www.microsoft.com/whdc/maintain/StartWER.mspx). Microsoft does not charge for this service.

To take advantage of WinQual, you must:

1. Register your company for WinQual, which requires a VeriSign ID. You can find Windows 7 information about WinQual in the developer portal grouped under Windows Vista SP1 \ Windows Server 2008. It will have a Windows 7 location soon.
2. Map the ISV applications to a product name and the ISV name, which links the failure reports to the company. Other ISVs cannot view your error reports.
3. Use the portal to identify top issues. ISVs can also create responses that inform customers what steps to take after a failure. The response system supports over 10 languages worldwide.

#### Links to Other Resources

**Debugging Tools for Windows:**

* **Overview**: <http://www.microsoft.com/whdc/DevTools/Debugging/default.mspx>

**Application Verifier:**

* **Overview**: <http://msdn.microsoft.com/en-us/library/ms644353.aspx>
* **Download**: <http://www.microsoft.com/downloads/details.aspx?FamilyID=bd02c19c-1250-433c-8c1b-2619bd93b3a2&DisplayLang=en>
* **For Microsoft Visual Studio 2008/.NET Framework 3.5**: <http://msdn.microsoft.com/en-us/library/ms220948.aspx>

**WinQual:**

* **Windows Quality Online Services (Winqual)**: <https://winqual.microsoft.com>
* **Windows Error Reporting: Getting Started**: <http://www.microsoft.com/whdc/maintain/StartWER.mspx>

### DISM Replaces pkgmgr, PEImg, and IntlConfg Tools

#### Affected Platforms

**Clients\*** - Windows Vista SP1 and later | Windows 7

**Servers** - Windows Server 2008 RTM and later | Windows Server 2008 R2

#### Description

The Deployment Image Servicing and Management (DISM) tool replaces the pkgmgr, PEImg, and IntlConfg tools that are being retired in Windows 7. DISM provides a single centralized tool for performing all of the functions of these three tools in a more efficient and standardized way, eliminating the source of many of the frustrations experienced by current users of these tools.

DISM includes a shim for Windows Vista SP1 and later as well as for Windows Server 2008 RTM and later, that redirects pkgmgr calls from legacy applications running on Windows 7 to DISM. If the application is running on one of the supported operating systems, the shim routes the call to pkgmgr.

No shims exist for legacy applications that call PEImg or IntlConfg.

#### Usage

DISM is transparent to pkgmgr end users on any of the supported platforms. However, if an application calls either PEImg or IntlConfg from Windows 7, the call will fail.

Update any scripts that call pkgmgr, PEImg, or IntlConfg to call DISM instead. It is important to include the updating of pkgmgr scripts in this effort, since the shim that provides backwards compatibility for pkgmgr will be removed in future versions of the Windows operating systems.

Check to ensure that calls to DISM have replaced any calls to pkgmgr, PEImg, and IntlConfg, and that the operation executes successfully.

### Compatibility - Windows Troubleshooting

#### Affected Platforms

**Clients** - Windows 7

**Servers** - Windows Server 2008 R2

#### Description

A new Windows 7 Solutions Center feature, Windows Troubleshooting, delivers a systematic troubleshooting experience for the user. The Solutions Center is one of the five icons pinned in the systray. Windows Troubleshooting allows you to browse or search for in-box troubleshooting packs and for troubleshooting packs that are stored on a Microsoft server on the Internet - a Better When Connected (BWC) experience. You can select and run a troubleshooting pack to attempt to try to resolve their problem. If you cannot identify a resolution to the problem, you have the option of searching help, community content, and support articles, or other troubleshooting packs for relevant related content. Should that not provide an answer to the problem, you can restore the PC to a time prior to when the problem occurred or get help through remote assistance. The intent is to allow you to find a solution to the problem easily and quickly.

#### Usage

The Solutions Center is clearly visible and available from several locations. You can launch it from the context menu of the Solutions Center in the systray, from the control panel as a shortcut link under system and maintenance, from the main page of the Solutions Center, and from Help content.

Troubleshooting packs are based upon powershell scripts. Microsoft is making public the process of authoring and publishing a troubleshooting pack to allow OEMs, IHVs, ISVs, and IT Professionals to develop and ship their own troubleshooting content.

For a consistent user experience, be sure to follow the best practices and guidelines described in the Windows troubleshooting toolkit when designing your own troubleshooting packs.

### Reliability Analysis Component

#### Affected Platforms

**Clients** - Windows Vista | Windows 7

**Servers** - Windows Server 2008 | Windows Server 2008 R2

#### Description

The Reliability Analysis Component (RAC) provides information to the Reliability Monitor that calculates the System Stability Index. The System Stability Index is a number from 1 (least stable) to 10 (most stable) and is a weighted measurement derived from the number of specified failures seen over a rolling historical period. Reliability Events in the System Stability Report describe the specific failures.

#### Usage

Use RAC's event content log and stability index to monitor the stability of your application. The API for the RAC data is WMI (Windows Management Instrumentation) where the data will be exposed via Win32\_ReliabilityStabilityMetrics and Win32\_ReliabilityRecords for Windows 7. (This data does not exist on download systems like Windows Vista). Use [Winqual](http://winqual.microsoft.com/) to determine the location of a crash.

#### Links to Other Resources

**Windows Quality Online Services (Winqual)**: <https://winqual.microsoft.com>

### Windows Error Reporting Problem Steps Recorder

#### Affected Platforms

**Clients** - Windows 7

**Servers** - Windows Server 2008 R2

#### Description

Prior to Windows 7, Watson could collect crash/hang reports indicating problems in need of repair. These reports contain information that is helpful in understanding the general nature of the problem, but not enough information to determine its root cause. For that, developers needed a tool to reproduce the crash/hang scenario for debugging. Watson did not have that capability.

We have added a new capability, Problem Steps Recorder (PSR), to Windows Error Reporting (WER) in Windows 7. This feature enables the collection of the actions that a user performed that led to a crash so that you can reproduce the situation for analysis and debugging.

#### Usage

This tool only affects the behavior of an application when a Windows Error Reporting service developer requests PSR enablement for that application. With PSR enabled for an application, the application may see some performance degradation. PSS/CSS plan to use this tool while troubleshooting with end user customers as well.

You must enable the PSR from the Watson backend.

### Terminal Services

#### Affected Platforms

**Servers** - Windows Server 2008 | Windows Server 2008 R2

#### Description

Terminal Services allows multiple concurrent users to access Windows Server in order to provide application and data hosting services using Microsoft “Presentation Virtualization” technology.

While most 32-bit and 64-bit applications run “as is” on Windows Terminal Services, several others do not perform as expected due to the difference in the platform (multi-user environment, concurrent access by multiple users, etc.).

For further information regarding application quality please read the whitepaper on [Application Readiness for TS](https://connect.microsoft.com/tsappcompat/content/content.aspx?ContentID=10015)

Visit the [Terminal Services product page](http://www.microsoft.com/windowsserver2008/en/us/ts-product-home.aspx) and the [TS TechNet](http://technet2.microsoft.com/windowsserver2008/en/servermanager/terminalservices.mspx) Web sites learn more about Terminal Services. To learn more about developing applications for Terminal Services, review the [TS Programming Guidelines](http://msdn.microsoft.com/en-us/library/aa383490(VS.85).aspx).

#### Manifestations of Impacts and Their Mitigations

Three changes in Windows 7 affect applications on Terminal Services:

* Windows Server 2008 R2 is 64-bit only
* Per-session IP Virtualization
* MSI-based deployments – User specific keys

##### 64-bit Only Windows Server 2008 R2

Applications written for 32-bit server will run in WoW mode and not natively on the Windows Server 2008 R2 or, hence, on Terminal Services. See the Windows 7 64-Bit Only topic for details.

##### *Mitigations for 64-bit only Win Server 2008 R2*

Most applications written for 32-bit will continue to work as normal in WoW mode. Any new applications written for Windows 7 Terminal Services should be developed and tested for deployment on 64-bit platforms.

##### IP Virtualization

Terminal Services allows per-app and per-session IP virtualization in Windows 7. IP virtualization enables applications/users that require a unique IP address, to work on Terminal servers.

**Per-app IP Virtualization:**

Per-app IP virtualization enables applications that require a unique IP address for each instance, to work on Terminal servers. For example, an application connecting to a database might be limited to one connection per IP address. Using per-app IP virtualization, multiple TS users can run this application on the TS server, and connect to the database.

**Per-session IP Virtualization:**

Per-session IP virtualization assigns a single unique IP address to all applications in a user session. This enables the user session to act as a unique network identity (that is, a single unique IP address for all applications in the user session).

##### *Mitigations for IP Virtualization:*

This feature improves application quality. As such, for applications that require a unique IP for each instance, the admin does not need to curb functionality on Terminal Services in Windows 7. To enable IP virtualization, the Admin goes to TS Configuration UI properties, clicks on the TS IP Virtualization tab, and checks the box marked “Enable IP Virtualization.”

##### MSI-based Deployments – User specific keys

MSI now takes care of installing user specific keys (HKCU) for all users on a Terminal Server. This allows better installation and deployment compatibility. This also makes the need for registry shadowing for installation redundant. MSI populates user specific keys on the first invocation after installation for the given user (that is, the user who is invoking the application).

##### *Mitigations for MSI based Deployments*

MSI-based deployments do not need to script creation of per-user keys for installation, as MSI takes care of this in Windows 7.

#### Links to Other Resources

* **TS Programming Guidelines:** [TS Programming Guidelines](http://msdn.microsoft.com/en-us/library/aa383490(VS.85).aspx)
* **TS at TechNet:** <http://technet.microsoft.com/en-us/ts/default.aspx>
* **Terminal Services product homepage**: [TS product homepage](http://www.microsoft.com/windowsserver2008/en/us/ts-product-home.aspx)
* **Application Readiness for Terminal Services whitepaper**: <https://connect.microsoft.com/tsappcompat/content/content.aspx?ContentID=10015>

### Network Hang Recovery

#### Affected Platforms

**Clients** - Windows 7

**Servers** - Windows Server 2008 R2

#### Description

When an executing application becomes unresponsive to user input, the application has "hung." When the application performs functions over a network, the application may become unresponsive to user input while waiting for a response from the network. This condition is referred to as a "network hang."

You can improve the reliability experience of a Windows application by providing the user with an opportunity to recover from a network hang. Windows' Network Hang Recovery breaks down the network connection used by the hung application. This event triggers the exception handler within the application that cancels pending network operations and may take other corrective action. Afterwards the application may be restored to a state in which it can respond to user input.

In some scenarios, the hang may be of a type such that it may not be recoverable without undesirable consequences. For example, when the hung application shares the network connection with another application, recovering a hung application by breaking down the network connection may interfere with the other application. Certain protocols may provide connections that can support multiple applications that would render the hang unrecoverable. Therefore, responding to a network hang includes diagnosing the type of hang; base this diagnosis on a protocol used in the network connection and/or whether other applications share the connection. Network Hang Recovery is offered only after it is determined that the hang is recoverable.

#### Usage

When Windows detects that a user is attempting to terminate an application experiencing a network hang, it offers the user an option to restore the application. Providing this option to the user may prevent the user from closing the application and thus losing unsaved data. It may also prevent other negative consequences of the user's terminating the application.

Be aware that a connection termination may be the result of an attempt to recover from a network hang.

Users can avoid Network Hang Recovery by *not* clicking on the Restore option when presented with this choice after they have chosen to close a hung application.

DO NOT make blocking calls, for example, calls to the network, in the UI thread; preserve responsiveness of the application

If using network connections, implement network exception handlers:

* Attempt to minimize disruptions by preserving users' data and state
* Avoid closing/terminating applications
* Avoid automatic retries without verification of connection state; retries may cause circular hangs

DO NOT depend on Network Hang Recovery to make your applications resilient to network hangs. Architect and program your application to be resilient to network hangs from the outset.

### Energy Efficiency Best Practices

#### Platform

**Clients** - Windows XP | Windows Vista | Windows 7

#### Description

Windows®-based laptops must meet energy efficiency regulatory requirements such as those of the United States Environmental Protection Agency (EPA) Energy Star program. Furthermore, surveys have shown that longer battery life continues to be what consumers most want and need in laptops.

To meet consumer demands, Windows laptops must continually advance in the following areas:

* Energy efficiency in all usage scenarios, including idle, productivity workloads, DVD and media playback, and industry benchmarks
* Mobile PC battery life - for hardware platforms and for Windows

The Windows platform is highly reliable and enables fast on-and-off performance. However, extensions provided with mobile PC systems, such as services, system tray applets, drivers, and other software, can significantly affect performance, reliability, and energy efficiency.

Energy efficiency is a complex problem, with factors affected by and affecting all elements of the PC ecosystem. Small improvements across multiple scenarios can improve energy efficiency, and yet a single poorly performing application, device, or system feature can increase energy consumption significantly.

Hardware and devices form the foundation for energy efficiency. However, application and service software must also be efficient to allow the system to achieve optimal battery life. Each software component on the system, including the operating system and value-added applications and services, must conform to basic efficiency guidelines. A single misbehaving application or service can eliminate any energy efficiency gains that the latest processor, devices, or platform hardware achieved. For more detailed information regarding battery life and energy efficiency please refer to the Battery Life Solutions Guide located on the Energy Efficiency Portal at <http://www.microsoft.com/whdc/system/pnppwr/mobilepwr.mspx>

The principle issues and components that affect battery life in a mobile PC are:

Battery Characteristics

* Size, type, and quality of battery capacity affect battery life
* The larger the battery, the greater the power supply
* Larger batteries are more expensive and heavier; users prefer lighter systems

Hardware Components

* Frequency and depth to which hardware can enter lower power states
* Hardware support of lower power states
* Driver optimization for energy efficiency

Operating System - Directed Power Management

* Efficiency of Windows code while under a load versus while idle
* Cooperation level of all components with Windows-directed power management
* Proper configuration of OS to optimize for power management through power policy settings

Application Software and Services

* Efficiency of applications, drivers, and services while under a load versus while idle
* Cooperation level of applications with Windows - directed power management
* Software allowance of the system or devices to enter into low-power idle states

A single application or service component can prevent a system from realizing optimal battery life. Although Windows provides many power configuration options, preinstalled software or power policy settings on many systems are not optimized for the host hardware platform.

A common method for evaluating the battery-life impact of preinstalled software is to compare the power consumption of the system with a clean installation of Windows versus a Windows installation that includes value-added software and services. Although a clean installation does not represent the typical platform that OEMs ship to customers, the power consumption comparison can provide insight into the energy-efficiency of preinstalled software.

#### Best Practices

To ensure that your application is optimized on Windows platforms, follow these best practices when you design applications or services:

* **Avoid use of high-resolution periodic timers**

Using high-resolution periodic timers (<10 ms) reduces the efficiency of processor power-management technologies.

* **Invest in performance optimizations**

Every performance optimization is a battery life optimization. Reductions in required resources, such as using less processor time or batching/clustering disk reads, allow system hardware to become idle and enter low-power modes.

* **Adjust to user power policy**

Windows Vista and later make it easy for the user to choose the overall power-savings or performance behavior of the system. Your application should respond to changes in power policy and reduce resource usage or increase performance accordingly. For example, an application should disable background activity such as indexing or system scanning when the user has selected a Power Saver power plan.

* **Reduce resource usage when the system is on battery power**

Your application should reduce its resource usage - such as background update frequency - when the system is on battery power.

* **Do not render to the display when it is off**

The system display might be off for power savings. Your application should not perform unnecessary graphics rendering when the display is off because this wastes system resources and power.

* **Avoid polling and spinning in tight loops**

Heavy processor usage reduces the effectiveness of processor power-management technologies such as processor idle states and processor performance states.

* **Do not prevent the system from turning off the display or idling to sleep**

Your application should make judicious power requests with the SetThreadExecutionState API. The system should make these requests only when critical operations must delay the system from powering off the display or automatically entering sleep.

* **Respond to common power-management events**

Your application should register for and respond to common power-management events, such as system power-source changes and power-on and power-off notifications for the display.

* D**o not enable debug logging by default; use Event Tracing for Windows instead**

Periodic debug logging can prevent disk spin-down.

#### Links to Other Resources

* **Battery Life Solutions Guide:** [http://www.microsoft.com/whdc/system/pnppwr/mobile\_bat.mspx#](http://www.microsoft.com/whdc/system/pnppwr/mobile_bat.mspx)
* **Energy Efficiency Portal:** <http://www.microsoft.com/whdc/system/pnppwr/mobilepwr.mspx>
* **Windows Performance Toolkit:** <http://www.microsoft.com/whdc/system/sysperf/perftools.mspx>

### Performance Improvement - Minimizing Unresponsive Services

#### Affected Platforms

**Clients** - Windows Vista | Windows 7

#### Description

Unresponsive services can result in timeouts, terminated sessions, and even lost data. Employing best practices can greatly reduce the occurrence of unresponsive services.

#### Best Practices

Make sure your applications and all of their dependent services and drivers respond to system power and shutdown notifications.

* All applications should respond promptly and appropriately to shutdown messages such as WM\_QUERYENDSESSION and WM\_ENDSESSION that indicate a shutdown is in progress.
* All services should promptly respond to SCM shutdown notifications. If they fail to do so, the machine treats them as unresponsive, initiates a 20-second timeout, and kills them opening up the possibility of lost data. This also adds 20 seconds to the shutdown time of a machine shutdown.
* All services that have kernel device-driver dependencies should respond promptly and appropriately to IRP\_MJ\_SHUTDOWN notification in their DispatchShutdown routines.

#### Links to Other Resources

**Windows Performance Toolkit:** <http://www.microsoft.com/whdc/system/sysperf/perftools.mspx>

### Certification for Windows 7 and Windows Server 2008 R2 Applications

#### Affected Platforms

**Clients** - Windows XP | Windows Vista | Windows 7

**Servers** - Windows Server 2003 | Windows Server 2008 | Windows Server 2008 R2

#### Description: Client Application Certification

The process for certifying applications for use of the Windows logo has changed more for client applications than for server applications. This topic discusses certification for each of these platforms.

In the Windows 7® timeframe, we want to continue building on the success of Windows and expand the logo program so that it is more relevant to customers. Based on partner and customer feedback, the Windows 7 logo program will focus on ensuring that devices, systems, and applications are compatible, reliable and can perform to meet Windows standards. The proposed program for Windows 7 will have only one logo for systems, devices, and applications.

We also improved the process for gaining certification. Microsoft will not require developers to have their applications tested for certification by authorized testing authorities. Instead, Microsoft is developing a test kit you can use to ensure your application meets the certification requirements. We have significantly reduced the number of requirements. Similar to the hardware program, we have created some policies; while not linked to specific test cases, we will enforce these policies. Partners must meet all requirements and observe all policies to gain Windows 7 certification.

Microsoft has focused the logo program on fundamentals that enable partners to innovate and reward them for their deeper technical investments through additional qualifications (AQs). AQs increase the visibility of key technologies by recognizing the investment and easily identifying them for customers. Information on the logo program, including AQs, will be available at <http://www.innovateonwindows.com/logo>.

#### Process: Client Application Certification

When applications are ready to be tested:

1. Establish a Winqual account at <https://winqual.microsoft.com/>
2. Create a VeriSign digital signature to ensure a secure transfer of your product information to Winqual at <https://securitycenter.verisign.com/celp/enroll/retail;jsessionid=HZpCyxsw5nz42yBnR36l1O1MJWT1wAgzXSS1rJ2fhwZsXOzdVOGJ!-254355561>
3. Follow the process as outlined by Winqual
4. Receive your certification

#### Description: Server Application Certification

The process for certifying server application in Windows 7 is similar to that for Windows Server 2008. There are five quality bars, or "pillars," for Logo certification:

* Windows Fundamentals
* Install and uninstall
* Security
* Reliability and high availability
* Hyper-V compatibility

Each quality pillar is divided into specific requirements that the application must meet. Requirements are further divided into deterministic tests.

Two Server Logo program tools are available to help test for the requirements, the Certification Tool and the Windows System State Analyzer.

#### Process: Server Application Certification

The Logo certification process includes both self-testing and a third party test. An authorized third party testing vendor performs most of the tests. However, some long-term and Cluster tests must be done by the developers themselves.

When you are ready to consider Logo certification:

1. Download the certification test tools from the Logo certification page (no registration or fee required).
2. Evaluate your applications against the technical bar.
   1. First level help for technical questions and clarifications is available at an [MSDN Forum](http://forums.microsoft.com/MSDN/ShowForum.aspx?ForumID=1850&SiteID=1).
   2. Second-level help is available via a dedicated mailbox at [WsLogoFB@Microsoft.com](mailto:WsLogoFB@Microsoft.com).
3. Certain self-test logs needs to be made available before Final certification is complete (for example, long-term stress tests are self-tested by developers; Microsoft only needs the logs).
4. When you are ready to submit your application for official testing, contact one of our authorized test vendors who will guide you from that point.
5. If all tests are successful, Microsoft will grant you the right to use the Certified For Windows Server logo on your packaging and in your advertising.
6. If any tests fail, Microsoft may assist you in resolving the failure, including evaluating waiver requests where applicable.

#### Links to Other Resources

* **Client Logo Certification for Applications**: <http://www.innovateonwindows.com/logo>
* **Server Logo Tools:** <http://www.InnovateOnWindowsServer.com>
* **Server Logo Certification for Applications**: <http://www.innovateon.com/>
* **Server Logo Program Certification Tool x86:** <http://go.microsoft.com/fwlink/?LinkID=111718>
* **Server Logo Program Certification Tool x64:** <http://go.microsoft.com/fwlink/?LinkID=111717>
* **Works With Tool:** <http://go.microsoft.com/fwlink/?LinkID=111715>
* **Windows System State Analyzer:** <http://go.microsoft.com/fwlink/?LinkID=111722>
* **MSDN Forum**: <http://forums.microsoft.com/MSDN/ShowForum.aspx?ForumID=1850&SiteID=1>
* **Logo Mailbox**: [SWLogo@Microsoft.com](mailto:SWLogo@Microsoft.com)
* **Windows Quality Online Services (Winqual)**: <https://winqual.microsoft.com>