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

**Version 0.9**

**October 28, 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**

[Introduction 5](#_Toc213041438)

[Application Quality Topics 7](#_Toc213041439)

[Internet Explorer 8—User Agent String 8](#_Toc213041440)

[Internet Explorer 8—Data Execution Protection/NX 9](#_Toc213041441)

[Removal of Windows Mail 11](#_Toc213041442)

[Microsoft Message Queuing (MSMQ)—Removal of Windows 2000 Client Support Service 13](#_Toc213041443)

[Compatibility—Operating System Versioning 14](#_Toc213041444)

[Server Core—WoW64 Is Now an Optional Feature 15](#_Toc213041445)

[User Interface—Enhanced Taskbar 16](#_Toc213041446)

[Microsoft Message Queuing (MSMQ)—Improved Queue Handling 20](#_Toc213041447)

[Windows Server — Terminal Services 21](#_Toc213041448)

[User Interface—High DPI Awareness 23](#_Toc213041449)

[Removal of WPDUSB.SYS Driver for Windows Portable Devices 25](#_Toc213041450)

[Server—Hyper-V 26](#_Toc213041451)

[Server—64-Bit Only 28](#_Toc213041452)

[File Library Replaces Document Folder 30](#_Toc213041453)

[New Binaries—Refactoring 31](#_Toc213041454)

[Compatibility—Application Manifest 32](#_Toc213041455)

[Microsoft Message Queuing (MSMQ)—SHA-2 Is the Default Encryption Algorithm 34](#_Toc213041456)

[User Interface—User Access Control Dialog Updates 35](#_Toc213041457)

[Tools, Resources, and Best Practices 36](#_Toc213041458)

[Application Verifier 38](#_Toc213041459)

[DISM Replaces pkgmgr, PEImg, and IntlConfg Tools 40](#_Toc213041460)

[Compatibility—Windows Troubleshooting 41](#_Toc213041461)

[Network Hang Recovery 42](#_Toc213041462)

[Reliability Analysis Component (RAC) 44](#_Toc213041463)

[Windows Error Reporting (WER) Problem Steps Recorder 45](#_Toc213041464)

[Certification for Windows 7 and Windows Server 2008 R2 Applications 46](#_Toc213041465)

# 

# 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, many new features have been introduced, existing features have been improved, and some features have been removed.

Among the enhancements to the operating system are an Enhanced Taskbar to improve user experience, and a Reliability Access Component to monitor application stability and isolate those sections of code that require Developer attention. 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. But more than that, it also points out differences in performance, reliability, and usability, and provides links to detailed white papers and other developer guidance.

Application Quality topics are organized from highest impact to lowest impact. This means that some similar topics (for example, those dealing with Microsoft Message Queuing) may not be grouped together.

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. Particularly useful are the:

* Application Verifier
* Problem Steps Recorder in Windows Error Reporting
* Reliability Analysis Component
* Network Hang Recovery
* Deployment Image Servicing and Management Tool

The *Tools, Resources, and Best Practices* section of this document provides information about each of these.

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

# **Application Quality Topics**

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. The topics are listed in the order of greatest impact and are hyperlinked to their location in the document:

* Internet Explorer 8—User Agent String
* Internet Explorer 8—Data Execution Protection/NX
* Removal of Windows Mail
* Microsoft Message Queuing (MSMQ)—Removal of Windows 2000 Client Support Service
* Compatibility—Operating System Versioning
* Server Core—WoW64 Is Now an Optional Feature
* User Interface—Enhanced Taskbar
* Microsoft Message Queuing (MSMQ)—Improved Queue Handling
* Windows Server — Terminal Services
* User Interface—High DPI Awareness
* Removal of WPDUSB.SYS Driver for Windows Portable Devices
* Server—Hyper-V
* Server—64-Bit Only
* File Library Replaces Document Folder
* New Binaries—Refactoring
* Compatibility—Application Manifest
* Microsoft Message Queuing (MSMQ)—SHA-2 Is the Default Encryption Algorithm
* User Interface—User Access Control Dialog Updates

# 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** – High

**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 IE User Agent String. Those that do so and depend on an earlier version number will be impacted. 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 Mode. See [Understanding User Agent Strings](http://msdn.microsoft.com/en-us/library/ms537503(VS.85).aspx) for details.

## Manifestation(s) of Impact

There are two impacted areas:

1. Web pages that explicitly check the User Agent String and do not support the IE8 User Agent String may not run properly. In the majority of cases, this means that users will be blocked from the content they are attempting to access or will receive incorrect / 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 IE7 using the Web Optional Component, but will not have access to IE8 features.

## Mitigations and Remedies

### 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 Mode for those applications based on IE7. 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 tests

Run your applications and Web pages in an IE8 environment on Vista or 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>

# 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 IE8.

## Feature Impact

**Severity** – High

**Frequency** – High

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 will typically 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, the DEP/NX feature is enabled for all Internet Explorer processes by default.

## Manifestation(s) 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.

## Mitigations and Remedies

### 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 (E.g. ATL) that are DEP compatible.

### Leveraging capabilities of feature

* 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 tests:

* Test your code with DEP/NX enabled using latest released IE version on Windows Vista SP1 or later.
* Alternatively, test with IE7 on Windows Vista after enabling the DEP/NX option. To enable DEP/NX for IE7: Run IE as an administrator, then set the appropriate checkbox in the Tools > Internet Options > Advanced tab

## Links to Other Resources

* **IE8 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>

# 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(s) 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 completely removed, others will fail when trying to launch.
* All DLLs ship in the box
* Publicly documented APIs continue to work as they did in 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

## Mitigations and Remedies

### Mitigation(s)

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

### Solution(s)

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 tests

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

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

## Platform

**Servers** – Windows Server 2008 R2

This change impacts Windows 2000 when interoperating in a Windows 7 domain where all domain controllers are 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 can be installed 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(s) of Impact

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.

## Mitigations and Remedies

### Mitigation(s)

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 capabilities of the feature

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 tests

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.

# Compatibility—Operating System Versioning

## Affected Platforms

**Clients** – Windows XP | Windows Vista | Windows 7

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

## Feature Impact

**Severity** – High

**Frequency** – High

## Description

The internal version number for Windows 7 and Windows Server 7 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(s) 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.

## Remedies

### Mitigation

Most applications will function properly on Windows 7 and Windows Server 7 because the application compatibility in Windows 7 and Windows Server 7 is very high. However, a Compatibility Mode is provided in Windows 7 and Windows Server 7 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 Mode from the Compatibility tab. In most cases, this should enable the application to operate properly without the need for any changes to the application.

### Solutions

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.

# Server Core—WoW64 Is Now an Optional Feature

## Platform

**Servers** – Windows Server 2008 R2

## Feature Impact

**Severity** – Medium

**Frequency** – High

## Description

WoW64 is no longer installed by default in the Server Core installation option for Windows Server 7. Instead, WoW64 is now an optional feature that can be installed if it is necessary to run 32-bit code.

In addition, the Active Directory, Active Directory Lightweight Directory Services, and Web Server roles require WoW64 to be installed in order to run in Windows Server 7.

## Manifestation(s) 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.

## Mitigations and Remedies

### Mitigation(s)

Install WoW64.

### Solution(s)

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 tests

Verify that all code used is 64-bit.

## 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>

# User Interface—Enhanced Taskbar

## Affected Platforms

**Clients** – Windows 7

**Servers** – Windows Server 2008 R2

## Feature Impact

**Severity** – Medium

**Frequency** – High

## Description

### General

Many of the new features offered on the Windows 7 Desktop are optional and are noted as such in the description. Please note that these features will become available with the Beta release, and will not be available with the M3 release.

By using the Windows 7 Taskbar, applications can provide more information to the user in more intuitive ways. Each application has a single button on the Taskbar, represented by a 32x32 icon. By default, each application’s running windows are all grouped under this single icon. In addition, the user can pin the application’s shortcut to the Taskbar to serve as an easily accessible application launcher. When the application is running, the button for this running application replaces the pinned launcher, so that the application always has a single, consistent location on the Taskbar. The application’s Taskbar button is the single place from which to access all the relevant information for this application, including the Destination List, Icon Overlays, Thumbnails for running windows, Thumbnail Toolbars, and Progress bars.

### Destination List

The Destinations List (name may change post-beta) helps users get to where they want to go faster. Examples of Destinations are files, URLs, tasks, or custom items that open within the application. The new Destination List within the Start menu and Taskbar makes common destinations and key tasks available via a single click. The Destination List is automatically populated based on frequency and recency of use for file-based applications. Additionally, an application can define custom destinations, enabling it to monitor its own destination usage and their semantics. Applications can also define Tasks (actions within the application that users will find convenient to access directly, for example, composing an e-mail) to appear in their menus.

### Icon Overlays

Icon Overlays used to convey long-standing status or notifications can be displayed on the bottom right-hand area of the application’s taskbar button. Icons are displayed one at a time, in the order in which they are received.

This feature is optional. Technically, both an icon image file (16x16 pixels) and an HWND are required.

### Thumbnails

Large, interactive thumbnails are provided by default to applications on hover. The Thumbnail API allows an application to define custom child windows and corresponding thumbnails for those windows. It also allows applications to pass a small or a full-size thumbnail for any of its windows.

This feature is optional.

### Thumbnail Toolbar

The Thumbnail Toolbar provides a single location for commonly performed key actions related to the window where the user does not need to restore the window, for example, Play/Stop controls for a media application.

This feature is optional. Technically, applications can provide a standard Win32 toolbar and get called back.

### Progress Bars

Applications will be able to show progress bars in their taskbar buttons so that users can stay aware of progress without having to keep the window visible. This will be useful for tracking file copies, downloads, installs, CD burning, or other operations with lengthy run times. Progress Bars can display all progress states supported by the progress common control: determinate, indeterminate, stopped, and paused.

This feature is optional. Technically, applications call the API by specifying a state and a value (0-100).

### Programmatic Placement of Gadgets

This feature allows ISVs to programmatically place a maximum of one gadget per application directly on the user’s desktop.

This feature is optional. Technically, the API is called during the application’s installation or first run.

## Manifestation(s) of Impact

### Deskband Usage

The new Windows Taskbar supports only the IDeskband2 class of deskbands that support Aero Glass. If an application uses a deskband, Aero Glass support is required.

### Quick Launch

Quick Launch will no longer be shown in the Taskbar. The underlying folder will still exist so as not to break applications’ installations, however, users will not see the Quick Launch bar on the Taskbar.

### Taskbar Buttons

The Windows 7 Taskbar will automatically try to connect the shortcut used to launch the application with all the running windows for this application and the application’s Destination List. If the Taskbar cannot successfully connect these parts of the application, it will have the following effects:

* Running windows that are part of the same application will not group together under a single Taskbar button
* The 2 commands in the Destination List for pinning the running application and launching a new instance will not work correctly
* Pinned shortcuts for this application will not be replaced by (that is, unify correctly) with the running windows for the application
* The Destination List will not show up correctly for the application

The common cases where these effects will manifest are:

* Applications that show UI from multiple executables
* Applications that show UI running in an executable that other applications run in as well (for example, host processes)

## Mitigations and Remedies

### Solution(s)

#### Deskband Usage

If an application uses a deskband, Aero Glass support is required.

#### Quick Launch

Remove any Quick Launch options from the application installer or other UI.

#### Shortcut Connections

The new AppUserModelIDs help the Taskbar connect an application’s shortcut to its windows on the taskbar. An application can override the Taskbar’s default grouping and unification process by declaring an explicit AppUserModelID.

Use an explicit AppUserModelID to connect all parts of the application. The same AppUserModelID must be on all components of a single application.

Different AppUserModelIDs can be used to separate the independent applications running in the same host process:

* Set the System.AppUserModel.ID as a property on all windows for this application OR use the API SetCurrentProcessExplicitAppUserModelID() to put the AppUserModelID on the application’s process(es) that show UI
* Set the PKEY\_AppUserModel\_ID property on all shortcuts that launch this application
* Set the AppUserModelID in progID registration for file-type registrations (if applicable)
* Use the AppUserModelID when calling the Custom Destination List APIs (if applicable)

### Leveraging capabilities of a new feature

#### Destination List

Customize Destinations, Categories, and Tasks by calling into the ICustomDestinationList API. ***Note***: While the feature name may change after Beta, the API name is final.

#### Icon Overlays

Set Icon Overlays by using methods in the ITaskbarList3 API.

#### Thumbnails and Thumbnail Toolbar

Set Thumbnails and Toolbars by using methods in the ITaskbarList3 API.

#### Progress Bars

Invoke Progress Bars by using methods in the ITaskbarList3 API.

#### Programmatic Placement of Gadgets

Apps can programmatically place a gadget on the user’s desktop by calling the RunGadget API during their installation.

### Compatibility, performance, reliability, and usability tests

While these features are not included in the builds available at the PDC, they will be available with the build refresh for Beta. Once you have updated with the Beta software, you can test that the application shows up correctly in the Windows 7 Taskbar by trying the following cases:

* Open multiple windows of the application (can be either multiple instances of the application or multiple windows within the application). Make sure all these windows appear grouped together under a single Taskbar button.
* While the application is running, open the Destination List in the Taskbar to see the 2 commands to pin this application to the Taskbar and launch a new instance (via the application’s shortcut). Make sure both of these commands work as expected. For example: Pin the application to the taskbar (from the Destination List while it’s running). Close the application, and make sure that the correct shortcut for the application is on the taskbar, and launching that shortcut opens the application with the expected experience.
* With the application pinned to the taskbar from step 2, try launching the application in other ways or from elsewhere. Make sure the resulting Taskbar button for the running window replaces the pinned launcher (that is, unifies and appears in the same place)
* If this is a file-based application, open a few files through Windows Explorer. Then make sure those files are being populated for this application’s Destination List.

# Microsoft Message Queuing (MSMQ)—Improved Queue Handling

## Platform

**Clients** – Windows 7

**Servers** – Windows Server 2008 R2

## Feature Impact

**Severity** – Medium

**Frequency** – Medium

## Description

MSMQ Service does not put a hard limit on the number of queues that can be created on a system. However, the performance of the system is impacted when a large number of queues is created. Specifically, when there are more than a few thousand queues, the start-up time of the MSMQ Service increases exponentially resulting in a visible impact.

Microsoft has optimized the MSMQ Service start-up in Windows 7 to reduce the look-up overhead for loading the queues into memory. This optimization has led to a dramatic improvement of the start-up time of the MSMQ Service even when several thousand queues are created in the system.

## Manifestation(s) of Impact

This performance improvement does not impact the functionality of any existing application.

## Mitigations and Remedies

### Leveraging the changed feature

Application developers using MSMQ on Windows 7 can now architect their solutions without limiting the number of queues. Note that the overall performance of the MSMQ Server is still affected by the number of queues, but the performance impact is now on a linear rather than exponential scale.

### Compatibility, performance, reliability, and usability tests

If you use a large number of queues, simulate the production environment on a test bed, monitor performance, and analyze the service start-up time and the message throughput with a large number of queues and messages present in the test system.

# Windows Server — Terminal Services

## Affected Platforms

**Servers** – Windows Server 2008 | Windows Server 2008 R2

## Feature Impact

**Severity** – Medium

**Frequency** -- Medium

## Description

Terminal Services (TS) 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 Compatibility on Terminal Services [*<<Link to be published>>*](link)

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 know 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 impact 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 Windows 7 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.

### Per session IP Virtualization

Terminal Services allows per-session IP virtualization in Windows 7. This allows applications that require a unique IP per instance to run in concurrent sessions on Terminal Services. This enables several applications that did not previously work with Terminal Services to work. This makes explicit checks for determining if the application is running in a Remote Desktop session to terminate or limit functionality of such an application unnecessary.

#### Mitigations for per session IP Virtualization

This is an application quality improvement feature. As such, the only change you need to make to applications is to enable compatibility of those that require a unique IP per instance. Such applications need not curb functionality on Terminal Services in Windows 7.

### MSI-based Deployments – User specific keys

Microsoft Windows Installer (MSI) now takes care of installing user specific keys (HKCU) for all users on a Remote Desktop 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 explicitly script for creating per-user keys for installation, as MSI will take care of this in Windows 7.

## Links to Other Resources

* **Terminal Services Programming Guidelines**: <http://msdn.microsoft.com/en-us/library/aa383490(VS.85).aspx>
* **Terminal Services at TechNet**: [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 Compatibility on Terminal Services whitepaper**: [*<<Link to be published>>*](link)

# 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.

Two new High DPI features have been added to Windows 7:

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

## Manifestation(s) of Impact

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

* 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

## Mitigations and Remedies

### Solution(s)

To make your applications DPI aware, follow these steps:

**Step 1**: Run the DPI Awareness Assessment and write down all the issues found

**Step 2**: Check each issue against the Common DPI Coding Issues

**Step 3**: Assess the cost of making the application fully DPI aware

**Step 4**: Make a list of the High DPI assets required (e.g., buttons, icons)

**Step 5**: Work through and fix the list of DPI issues found in Step 1

**Step 6**: Integrate the new assets from Step 4

**Step 7**: Declare your application DPI Aware

**Step 8**: Re-run the DPI Awareness assessment and verify the issues are fixed

### Compatibility, performance, reliability, and usability tests

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

# 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; support of these have also been removed.

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

## Manifestation(s) of Impact

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

## Mitigations and Remedies

### Mitigation(s)

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(s) for Impact to an existing feature

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 (e.g., 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 capabilities of a new feature to resolve Impact

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>

# Server—Hyper-V

## Affected Platforms

**Clients** – Windows XP | Windows Vista | Windows 7

**Servers** – Windows Server 2008 | Windows Server 2008 R2

## Feature Impact

**Severity** – Low

**Frequency** – Low

## Description

Server virtualization enables multiple operating systems to run on a single physical machine as virtual machines (VMs), allowing you to consolidate workloads of underutilized server machines onto a smaller number of fully utilized machines. Windows 7 includes several enhancements to the Windows Server 2008 version:

* **Live Migration:** In Windows Server 2008, we had Quick Migration. With Live Migration, we improved migration speed and storage flexibility.
* **Dynamic Memory:** When allocating memory to a Virtual Machine, you can now configure minimum and maximum memory in order to allow the virtual machine to dynamically grow and shrink its memory usage.
* **Logical Processor Support**: We increased logical host processors from 16LP to 32LP.
* **Storage Hot Add:** Now you can add additional VHD or Pass-through disks to a running virtual machine without turning off the virtual machine.
* **New Hardware Support:** We have added support for the technologies included in new processors and network cards coming to market, including Second Level Translation (SLAT), TCP Offload (Chimney), and VMdQ.
* **Terminal Services virtualization (TSv):** We centralized desktop solution for Hyper-V.

## Manifestation(s) of Impact

* **Live Migration**: You may need to change the way you have architected your storage systems in order to fully leverage this technology. Though these changes may not be required, you may choose to implement them in order to fully leverage the benefits. You may need a management application to orchestrate Live Migration.
* **Dynamic Memory**: Upon migrating to Windows Server 2008 R2, this feature will have no immediate impact on existing virtual machines. You can choose not to enable this feature. Management applications that configure the settings of a virtual machine may require updating in order to manage this new feature.
* **Logical Processor Support**: This feature will have no impact on customers or ISVs when migrating from Windows Server 2008 to Windows Server 2008 R2.
* **Storage Hot Add**: This feature will have no impact to customers or ISVs when migrating from Windows Server 2008 to Windows Server 2008 R2. Management applications that configure the settings of a virtual machine may require updating in order to manage this new feature.
* **New Hardware Support**: These features apply only to new hardware that is being introduced to the market. Because it will not have build-in support for these features, it is not likely that a physical server that is being migrated from Window Server 2008 to Windows Server 2008 R2 will be impacted. If these features are available on the server that is being migrated, no direct changes are anticipated.
* **Terminal Services virtualization**: This feature will have no impact to customers or ISVs when migrating from Windows Server 2008 to Windows Server 2008 R2. Applications that leverage Terminal Services (TS) may be impacted. This feature directly integrates with TS and therefore applications that configure TS may require updating in order to manage this new feature.

## Mitigations and Remedies

### Mitigation(s)

* **Live Migration:** Provide prescriptive Guidance to end users with best practices and recommendations for storage system design. You must inform the end user about the options and recommendations.

### Leveraging capabilities:

* **Live Migration:** This feature enables the Dynamic IT environment. Virtualization Management Application Developers must modify their application to leverage this new feature. Microsoft will make WMI interfaces publically available to allow a developer to integrate applications with this feature.
* **Dynamic Memory:** This feature enables greater consolidation of Virtual Machines on Hyper-V. Virtualization Management Application Developers must modify their application to leverage this new feature. Microsoft will make WMI interfaces publically available to allow a developer to integrate applications with this feature.
* **Terminal Services virtualization**: This feature enables a new virtualization scenario. Applications that leverage Terminal Services (TS) may be impacted. This feature directly integrates with TS and therefore applications that configure TS may require updating in order to manage this new feature.

### Compatibility, performance, reliability, and usability tests as appropriate:

None of these new features should have negative performance, reliability, or usability impacts.

## Links to Other Resources

**WMI management interfaces for Hyper-V v1** are available at <http://msdn2.microsoft.com/en-us/library/cc136992(VS.85).aspx>. While most of this content will apply to v2 of Hyper-V, an updated version with v2-specific information should be available closer to the Windows 7 launch.

# Server—64-Bit Only

## Affected Platforms

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

## Feature Impact

**Severity** – Medium

## Description

Windows Server 7 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(s) of Impact

This will impact three areas:

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

## Remedies

### 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. Console, GUI, and service applications are supported. 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>

# 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 will be 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. Libraries can be discovered using the Known Folder system (for example, FOLDEDID\_DocumentsLibrary).

## Manifestation(s) 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.

## Mitigations and Remedies

### Solution for change to an existing feature impact

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 capabilities of a new feature

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 they can be enumerated just like any other Shell Folder.

### Compatibility, performance, reliability, and usability tests

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

## Links to Other Resources

MSDN documentation for IShellLibrary

# New Binaries—Refactoring

## Affected Platforms

**Clients** – Windows 7

**Servers** – Windows Server 2008 R2

## Feature Impact

**Severity** – Medium

**Frequency** – High

## Description

Functionality has been relocated to new low-level binaries to improve internal engineering efficiencies and improve foundation for future work. 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(s) of Impact

Functionality has been relocated to new low-level binaries. As an example of this, 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).

## Mitigations and Remedies

### Solution(s) for Impacts to an existing feature

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.

# 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.

The current list of Windows components that support this legacy behavior in Windows 7 are:

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

Note that this manifest section will have no effect on Windows XP and Windows Vista.

## Remedies

### Leveraging new capabilities to resolve legacy behavior:

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 tests as appropriate:

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

## Links to Other Resources

**Logo Certification for Applications**: <http://www.innovateon.com/>

# Microsoft Message Queuing (MSMQ)—SHA-2 Is the Default Encryption 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, all incoming messages must be signed with SHA-2. Support for a lower encryption algorithm can be enabled through an administrator-accessible registry key.

## Manifestation(s) 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.

## Mitigations and Remedies

### Mitigation(s)

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.

# User Interface—User Access Control Dialog Updates

## Affected Platforms

**Clients** – Windows 7

**Servers** – Windows Server 2008 R2

## Feature Impact

**Severity** – Low

**Frequency –** The frequency of running into the UAC dialogs depends on the application. The dialog comes up whenever the application needs to run elevated.

## Description

In Windows 7, the UAC dialog box choices have been standardized. 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.

## Mitigations and Remedies

### Leveraging the Updates to User Access Control Dialogs

The application development requirements in Windows 7 for UAC compatibility are the same as in Vista. Vista-compatible applications will interact with UAC in Windows 7 without any modifications. See the User Access 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>

# **Tools, Resources, and Best Practices**

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. The available tools and resources include the following items, which are hyperlinked to their locations in the document:

* Application Verifier
* DISM Replaces pkgmgr, PEImg, and IntlConfg Tools
* Compatibility—Windows Troubleshooting
* Network Hang Recovery
* Reliability Analysis Component (RAC)
* Certification for Windows 7 and Windows Server 2008 R2 Applications

# Application Verifier

## Affected Platforms

**Clients** – Windows XP | Windows Vista | Windows 7

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

## Feature Impact

Only those who previously had undetected errors in their code will notice any changes.

**Severity** – High

**Frequency** – Low

The severity is “High,” as we have inserted a crash or break-point when an errant condition is discovered; frequency is expected to be “Low,” as the number of new crashes detected is not expected to be overwhelming.

## Description

Application Verifier should be promoted and enforced as a quality gate for all development. Several improvements have been made to it:

* We have provided additional checks to address issues that were discovered by the Windows Error Reporting team in 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.

This feature is typically run under Debug versions only, so performance is not typically 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 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 | Windows 7

**Servers** – Windows Server 2008 | Windows Server 2008 R2

\* Note: Supports Vista SP1 and later, and Server 2008 RTM and later

## Feature Impact

**Severity** – Low

**Frequency** – High

## 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

## Feature Impact

**Severity** – Low

**Frequency** – Low

## 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. The process of authoring and publishing a troubleshooting pack will be made public 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.

# Network Hang Recovery

## Affected Platforms

**Clients** – Windows 7

**Servers** – Windows Server 2008 R2

## Feature Impact

**Severity** – Low

**Frequency** – Low

## Description

When an executing application becomes unresponsive to user input, the application is said to be "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.”

The reliability experience of a Windows application may be improved by providing the user with an opportunity to recover from a network hang. When Windows detects a user’s attempt to terminate an application experiencing a network hang, the user is offered 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. Other negative consequences of the user’s terminating the application may also be prevented.

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; this diagnosis may be based on a protocol used in the network connection and/or whether the connection is shared with other applications. 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, the user is offered 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. Other negative consequences of the user’s terminating the application may also be prevented.

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. Applications should be architecture and programmed to be resilient to network hangs from the outset.

## Links to Other Resources

None.

# Reliability Analysis Component (RAC)

## Affected Platforms

**Clients** – Windows Vista | Windows 7

**Servers** – Windows Server 2008 | Windows Server 2008 R2

## Feature Impact

RAC is a tool that can help you locate sources of problems in your code. Therefore, impact is not relevant.

## Description

The Reliability Analysis Component 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 (WER) Problem Steps Recorder

## Affected Platforms

**Clients** – Windows 7

**Servers** – Windows Server 2008 R2

## Feature Impact

**Severity** – Low

**Frequency** – Low

## 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.

A new capability, Problem Steps Recorder (PSR), was added to Windows Error Reporting 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 impacts the behavior of an application when a Windows Error Reporting service developer requests that the PSR be enabled for that application. When PSR is 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.

## Links to Other Resources

None.

# 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

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.

## Description: Client Application Certification

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.

In addition, the process for gaining certification has been improved. 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, these policies will be enforced. All requirements must be met and all policies must be observed to gain Windows 7 certification.

Microsoft has focused the logo program on fundamentals that will enable the partners to innovate and be rewarded 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:

<https://winqual.microsoft.com/>

1. Create a VeriSign digital signature to ensure a secure transfer of your product information to Winqual:

<https://securitycenter.verisign.com/celp/enroll/retail;jsessionid=HZpCyxsw5nz42yBnR36l1O1MJWT1wAgzXSS1rJ2fhwZsXOzdVOGJ!-254355561>

1. Follow the process as outlined by Winqual
2. 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:

* Certification Tool
* 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:** [www.InnovateOnWindowsServer.com](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>