Reliability, Availability and Serviceability Enhancements in Windows Server 2008

Microsoft Corporation

Published:

Project Author: Enrique Saggese

Project Editor:

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

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

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

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

 (2008) Microsoft Corporation. All rights reserved.

Microsoft, Windows Server, Windows NT, Windows Server, Windows Vista, SQL Server, Active Directory, Windows and Hyper-V are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

# Abstract

In Microsoft® Windows Server 2008, Microsoft is introducing many new features and technologies that will help to provide enhanced reliability, availability and serviceability to Windows Server based solutions. This document describes some of these features and technologies.

Contents

[Introduction 1](#_Toc194830975)

[System Availability vs. Service Availability 4](#_Toc194830977)

[R/A/S enhancements in core Windows Server 2008 platform 4](#_Toc194830978)

[Reliability enhancements in Windows Server 2008 6](#_Toc194830979)

[Kernel Patch Protection 6](#_Toc194830980)

[Transactional File System 7](#_Toc194830981)

[Windows Hardware Error Architecture 7](#_Toc194830982)

[User Account Control 9](#_Toc194830983)

[Service Hardening 10](#_Toc194830984)

[New availability Features in Windows Server 2008 12](#_Toc194830985)

[Kernel Mode Code Signing 13](#_Toc194830986)

[Self Healing NTFS 15](#_Toc194830987)

[Failover Clustering enhancements 15](#_Toc194830988)

[Serviceability improvements in Windows Server 2008 16](#_Toc194830989)

[Reliability and performance monitoring 17](#_Toc194830990)

[Windows Eventing 6.0 19](#_Toc194830991)

[Hot Patching 19](#_Toc194830992)

[Server Core 20](#_Toc194830993)

[Internet Information Services 7 22](#_Toc194830994)

[Hot Addition and Replacement of CPU, Memory, ACPI buses and cards 23](#_Toc194830995)

[Virtualization Technologies impact on R/A/S 24](#_Toc194830996)

[Other R/A/S enhancements 25](#_Toc194830997)

[Definition of Failure modes and Windows Server 2008 Enhancement Impact 27](#_Toc194830998)

[Appendix A: System Availability Calculation 28](#_Toc194830999)

[Definitions 33](#_Toc194831000)

[Examples 34](#_Toc194831001)

# Introduction

Windows has come a long way since the times of Windows NT 4.0 Server, and this is particularly visible in the operating system stability, reliability, and high availability and serviceability capabilities. Changes in architecture, code improvements and new features and capabilities in the operating system have made the operating system more reliable and serviceable, and capable of implementing highly available solutions.

At the same time, these new capabilities have enabled Windows-based solutions that are becoming more business-critical. As these solutions grow in complexity, and companies become more and more reliant on their Windows systems, the resulting needs pose higher demands on the operating system. The new expectations make maximum availability and reliability a very appealing characteristic. In the same fashion, the ability of a system to be serviced and withstand hardware, management, and software errors without disrupting operations becomes a business enabler.

Windows Server 2008 implements a significant number of new features and capabilities that provide an increased ability to operate in critical environments. In this document, we discuss those enhancements and how they impact the Reliability, Availability and Serviceability of Windows based IT services.

# Reliability, Availability and Serviceability definitions

Reliability, Availability and Serviceability, often referred to collectively as R/A/S, is a term coined in the eighties to encompass the capabilities of a computer system that allow it to run in a predictable fashion and without interruptions. The three terms are used to refer to varying meanings, so we will define them as follows in the context of this document:

**Reliability** is formally defined as a measure of the time between failures (as a numeric value such as a mean time between failures), where failure is defined as a departure from acceptable service for an application, a computer system, or the network system. It is a measure of how consistently a machine or set of machines perform the tasks that are their purpose. For a system to be reliable, the integrity of the data processing as well as its transmission and storage both in temporary and permanent memory is extremely important. A data storage or processing system such as a database server which allows even a small percentage of corruption to occur might cause severe adverse effects on business functions. The ability to keep data without unintended alterations is the main feature in this regard. The ability to detect such alterations and, if possible, repair such data to its original state is also a very important. In any large system with a high amount of data, at some point, after a certain time, some type of error is bound to occur. Some techniques normally used to this end are data integrity checks, parities and redundant storage of data, correction algorithms and the ability to perform resource lock management, which prevents access to resources during modification. The ability to deliver results in the expected time and form is also considered part of its reliability.

**Availability** is the probability that a system is in its intended functional condition and therefore capable of being used in a stated environment. It measures the level of service provided by applications, services, or systems. Highly available systems have minimal downtime, whether planned or unplanned. Availability is often expressed as the percentage of time that a service or system is available, for example, 99.9 percent for a service that is down for 8.75 hours a year.

In contrast to reliability, availability also involves the time required to bring a system back to normal operations after it fails or when it is taken down for planned maintenance or upgrade.

**Serviceability** is a measure of the ability of a system to monitor failures or anomalies, to report actual or potential failures in time for proactive or corrective action. Another key measure is the ability to correct the failures without service interruption.

**Failure** is any event that impacts a system or service in a way that adversely affects the system performance criteria. For example, the criteria could include, in addition to operational status, specific output in a sold-out condition, maintenance cost, resource consumption, or safety requirements, etc. A failure specification should contain specific criteria and not be ambiguous. Failure definition can change on a given system over time.

**Failure Rate** is the number of failures of an item per unit measurement of life. Failure rate is considered constant over the useful life period.

**Mean Time Between Failures** (MTBF) is the total operating time divided by the number of failures. In other words, MTBF is the sum of the Mean Time to Failure (MTTF) and Mean Time to Restore (MTTR). MTTF can be calculated as the inverse of failure rate.

**Mean Time To Restore** (MTTR) is the total elapsed time from initial failure to the reinitiating of system status. Mean Time To Restore includes Mean Time To Repair (MTBF + MTTR = 1.)

There are several availability classification schemes, some based on specific customer requirement. The following availability classification has been commonly used in the market for many years.

|  |  |  |
| --- | --- | --- |
| **Availability Classification** | **Availability %** | **Downtime Per Year** |
| Continuous | 100.0% | 0 |
| Fault Tolerant | 99.999% | 5 minutes |
| Fault Resilient Clusters with Failover | 99.99% | 53 minutes |
| High Availability | 99.9% | 8.3 hours |
| Commercial Availability | 99.5% | 43.8 hours  |

As mentioned in the introduction to this document the demand for commercial availability has gone up recently due to the increased dependency of businesses on their information systems. Also, the typical availability of individual servers and clusters has improved with recent technological advances.

# System Availability vs. Service Availability

Even though availability includes observation of individual system results, it is actually the end result to the users- the whole service availability- that matters most.

For small or less critical systems that implement no redundant solutions individual system reliability is paramount, as it dictates the whole service reliability. Even for systems incorporating high availability solutions such as failover clustering or load balancing individual server availability is still relevant since it indicates how seamless the operation of a platform is and how many times the high availability solution enters into play. In such systems where high availability measures are in place, individual server reliability is an important factor in cases such as:

* When individual server reliability is so low that there is a significant chance of simultaneous multiple server failure. For example, if each individual server of a two server group implementing some sort of load balance or failover has an individual availability of 90%, that means the total service availability could be close to 99%, which implies about three days of downtime a year for the total service, so any improvement on individual server availability will have an impact on service availability.
* When a failure occurs at a “Single Point of Failure”, a scenario where there is some sort of systemic failure that, under the same circumstances, would make every server in the farm fail. That can be general power or communications failures, management failures or data/event related software failures. In those cases, it is likely that a specific type of failure would happen on every system at the same time, bringing the whole service down regardless of redundancies.
* When individual server failures have a related management, operations or support cost. In those cases, each system’s failure has a cost regardless of the end service being operational.

For complex and redundant systems, there are several considerations that need to be taken in account when evaluating service availability. In , we include an analysis of such considerations.

In this paper we mainly address enhancements that benefit individual server availability. Specific exceptions to this are the enhancements in clustering services and virtualization technologies that, even though they can enhance each individual systems operation, aim at enhancing the total service uptime.

For reliability and serviceability, these considerations cannot be applied generally to every situation. If a single system is providing unreliable data, the most common consequence is that the whole end service provides unreliable results regardless of fail over or load balancing solutions that might be in place. Serviceability capabilities such as early failure detection or online problem resolution, apply individually to each server regardless of the architecture in which they are implemented.

# R/A/S enhancements in core Windows Server 2008 platform

Three of the main focus areas for the design of Windows Server 2008 were to improve reliability, availability and serviceability of the platform. Even though Windows Server 2003 has had excellent results in this area, the criticality of Windows based solutions has gone up with time. In critical and many non critical systems, events such as downtime, service work, and errors equal lost money. The constant focus on cost optimization implies that every optimization in this area counts.

The first enhancement to examine in Windows Server 2008 is in general code quality. The work done for Windows Vista and Windows Server 2008 included a general rewrite of many areas of the product, a complete review of the code base according to the new Common Engineering Criteria and lots of improvements based on the requirements of the Trusted Computing initiative. These enhancements include significant changes in architecture, clean rewrite of many APIs and code improvements throughout the products. The result observed in Windows Vista is a significantly more reliable product, with both the architectural and coding enhancements showing big promise.

One of the disadvantages of such a significant rewrite is that highly reviewed and tested old code is replaced with new code that hasn’t been exposed to the general community for years of review, and in any new development it is likely that new bugs will be introduced. The new code, on the other hand, was written from the start by following new coding standards. The standards are mandated by the new secure coding practices developed for the Trusted Computing Initiative, and all the design factors established for the Common Engineering Criteria. The new coding practices, together with the very intensive code review and testing performed for each new or revised piece of code, led to high quality coding which has shown to be very robust from the initial launch with a lower bug report rate than with similar products in the past.

Despite all this, the quality and robustness requirements for a server product are considerably higher than for a desktop operating system (OS). To meet this requirement it was decided that the fundamental components of Windows Server 2008 would be based not on the common codebase with Windows Vista, but on Windows Vista Service Pack 1. This implies that Windows Server 2008 will benefit from all the fixes and codebase improvements developed after the release of Windows Vista for Service Pack 1 for the common components in both operating systems.

Another benefit is, since the driver model for Windows Server 2008 is shared with that of Windows Vista, the drivers for the OS provided from Microsoft and from third parties, will be more widely available, more mature and better tested at release than what could be expected for a new OS using a driver model for which little experience existed in the market.

What the general maturity of the code and the driver model imply is that it can be expected that the OS will be highly mature and reliable from the day it is released.

There are specific improvements in the OS that affect Reliability, Availability and Serviceability of the OS. The improvements, some shared with Windows Vista and others specific to Windows Server 2008, are addressed individually in the following sections. While we are introducing them in separate sections for reliability, availability and serviceability, it must be noted that most of the features mentioned affect more than one of these areas.

# Reliability enhancements in Windows Server 2008

## Kernel Patch Protection

### What is Kernel Patch Protection

Originally introduced for Windows XP Service Pack 2 x64 and Windows Server 2003 SP1 x64, Kernel Patch Protection (previously known as Patchguard) prevents malware and non standard applications from patching the kernel.

"Kernel patching" or "kernel hooking" is the practice of using unsupported mechanisms to modify or replace kernel code or kernel data tables. Patching fundamentally violates the integrity of the Windows kernel and is undocumented, unsupported and has always been discouraged by Microsoft. Kernel patching can result in unpredictable behavior, system instability and performance problems—like the Blue Screen of Death–which can lead to lost user productivity and data. More importantly, kernel patching has increasingly become a mechanism used by malware developers to attack Windows systems.

### How does it work

Kernel Patch Protection is a mechanism used by the Windows x64 kernel to prevent applications from patching the kernel.

Specifically, Kernel Patch Protection will stop an attempt of performing any the following operations:

* Modifying System service tables
* Altering the Interrupt Descriptor Table (IDT)
* Modifying the Global Descriptor Table (GDT)
* Generating kernel stacks not allocated by the kernel
* Patching any part of the kernel code

When any of these operations is attempted, even if it is done by a privileged process, the OS will block the attempt, halting the system to guarantee integrity.

### What is the impact on Windows Server 2008 R/A/S

Kernel Patching is frequently used by two kinds of applications. The first one is malware. Virus writers and rootkit writers often patch the kernel to enable intercepting OS level calls in order both to hide them from inspection and to be able to sniff user mode activity for their own purposes.

The other kind of application that frequently patches the kernel is the anti malware tool. They often patch the kernel in unsupported ways in order to detect and intercept malware attempting to patch the kernel and install its code, to be able to scan the system at a low level and to be able to sniff suspicious activity in kernel calls.

Even though this later type of usage is well intentioned, it is often the case that it ends up damaging the systems stability and reliability. Patching the kernel often has detrimental effects on performance, and untested scenarios are likely to fail when attempted by the user. One such significant case happened with the introduction of clustering to the Windows platform. Many antivirus applications use kernel patching that used to work well in standalone servers but broke when applied to clusters, and many continued to work but caused significant problems to the clusters operations. At one point, a significant percentage of cluster support calls to Microsoft were related to antivirus installations.

Another big issue is that while Microsoft is committed to maintain API compatibility from one OS release to another, the internal structure and implementation of the kernel libraries might change significantly between releases. This often leads to security applications breaking when moving to new versions of the OS, and potentially when installing individual patches.

Finally, when different applications attempt kernel patching on the same system, the end results can be unpredictable.

The net result is that kernel patching significantly affects the OS’s reliability, (such as when done by a virus), and availability (when done by administrative applications such as antivirus tools) and should be avoided.

By preventing kernel patching, Windows Server 2008 helps insure system integrity. It will block certain types of malware that could affect the system reliability and integrity. It also helps prevent damage by well intentioned applications that might attempt to patch the kernel for management tasks such as virus scanning, with potential reliability effects due to the nonstandard and unsupported type of operation.

Legitimate low level hooking operations should be done instead thorough the appropriate published APIs. By using the official APIs to do such hooking applications can perform their duties without compromising the systems availability and reliability. Microsoft has worked with malware protection software vendors to ensure the proper interfaces necessary for monitoring and repairing the system at a low level are available and correctly used by the software.

It is important to consider that Kernel Patch Protection is only available on the 64 bit versions (x64) of Windows Server 2008. It is expected that the benefits of proper application behavior will in many cases be extended to 32 bit versions of the same tools when the application developers update their tools to work on 64 bit versions of the OS.

## Transactional File System

### What is Transactional File System and Transactional Registry

First introduced in Windows Vista, Transactional File System and Transactional Registry are enhancements done to the NTFS file system and Windows Registry that enable the definition of ACID (Atomic, Consistent, Isolated and Durable) transactions for file operations and registry operations. The new functionality includes the new file system and registry transactional capabilities, the related API calls and some tools that allow for easy usage within scripts.

### How does it work

The transactional file system and transactional registry work by allowing the definition of transactions that encompass one or multiple operations. In a typical operation, the beginning of a transaction is defined, a series of operations are executed that produce changes in files, the registry or other components, and then transaction is committed. If any of the operations fail, for example, because one service or server becomes unavailable or because the server where the data is being written crashes, the whole set of operations is rolled back to the initial state immediately in the case of operation cancellations. The transaction is rolled back on the next boot in situations that result in server failures. This behavior guarantees that the set of operations is executed as an integral unit, with either all the operations or no operations executed. It also allows for consistent visibility to other applications of the data, so any access to the data made by other applications see all the data in the initial state until the transaction is committed.

The Transactional File System and the Transactional Registry work with the Kernel Transaction Manager, the Common Log File System and the Distributed Transaction Coordinator to provide integral transaction capability encompassing the local file system, remote file access, the registry, databases and applications.

### What is the impact on Windows Server 2008 R/A/S

Transactions guarantee consistency, both within operations and between operations.

A single file modification done within a transaction will be either complete or rolled back, guaranteeing the file to be always in a consistent state. Without a transaction, the file could become inconsistent if an operation is cancelled midstream, thus resulting in reliability issues.

For multiple file operations, transactions guarantee consistency among the data between separate files. For instance, a single transaction could make two related modifications in separate files (such as reducing a number on one of the files and increasing a number in another file in a similar amount) and if the transaction is aborted both files would return to the initial, consistent state. Without a transaction, one of the files would end up modified and the other would not, leaving data in an inconsistent state. Whole file operations can also be transacted, such as moving a group of files from one folder to another. With a transaction either all or none of the files end up moved

Transactions can also protect registry operations. Several registry changes are grouped in a single transaction, and either all of them are committed or all of them return to the original state. Registry operations can be combined with other operations such as file system operations in a single transaction.

Finally, transactions can protect related data in heterogeneous sources. For instance a database could contain pointers to files in a folder or files in a remote share. When done within a transaction, the deletion or addition of a pointer can be made together with the deletion or addition of the corresponding file, guaranteeing that invalid pointers or orphan files will never be left because of an operation interruption.

Transactional File System and Transactional Registry protect data integrity in case of operation interruption and provide an always consistent view of data- even during operations.

## Windows Hardware Error Architecture

### What is Windows Hardware Error Architecture

In versions of Windows Server previous to Windows Server 2008 the OS supported several unrelated mechanisms for reporting hardware errors. These mechanisms provided little support for error recovery, and for uncorrected errors, the OS simply halted the system. Next, during a subsequent session, the system recorded some of the available error information in the system event log. This implementation resulted in operation interruptions and made it difficult to troubleshoot errors due to lack of detailed information.

Windows Hardware Error Architecture, or WHEA, is a common OS level hardware error handling architecture implemented in Windows Vista and Windows Server 2008 that allows for consistent error source discovery, recording and handling. WHEA takes advantage of the additional hardware error information available in today's hardware devices and integrates much more closely with the system firmware.

### How does it work

WHEA includes a series of components that handle the flow of errors from the hardware and firmware to the OS. The components also implement the appropriate actions depending on the error source and type.

For each hardware error source reported by the platform to the OS, WHEA uses a corresponding Low Level Hardware Error Handler (LLHEH), which is the first OS code that executes in response to a hardware error condition. The LLHEH can be an interrupt handler, an exception handler, polling routine, or a callback routine that is invoked by the system firmware.

LLHEH generally perform the following tasks:

• Acknowledge the hardware error.

• Capture the available error information related to the hardware error.

• Report the hardware error condition to the OS.

LLHEHs either interact directly with the hardware and firmware to retrieve hardware error information and build an error packet or retrieve the error packet prebuilt by the firmware.

All LLHEHs report hardware errors to the Windows OS by passing the hardware error packet data to a common error reporting interface. WHEA creates an error record in a standardized format from the above error packet.

The platform-specific hardware error driver (PSHED) is a component that provides an abstraction of the hardware error reporting facilities of the underlying platform. This allows the core Windows components to access only the error status registers that are considered to be architectural, while also providing a mechanism through which richer and more detailed platform-specific hardware error information can be obtained. PSHEDs for each supported processor architecture (Itanium, x64, and x86) are provided with the OS, but platform vendors can supplement the default PSHED functionality by implementing PSHED plug-in modules that take advantage of platform-specific capabilities.

Upon notification of a hardware error condition, Windows constructs a hardware error record that describes the hardware error condition. Windows then calls into the PSHED so that it can add any additional information to the hardware error record to better describe the error condition. After Windows has compiled all of the hardware error information into the error record, WHEA attempts error recovery if this was a correctable error, and it notifies user-mode applications by generating a Windows Event Tracing event.

Under certain hardware error conditions, the OS is forced to restart the computer to recover from the error. In these situations, since the interruption is immediate, the OS does not log the error information in the system event log or notify user-mode applications until after the computer has been restarted. Therefore, the OS must save the hardware error record to some form of non-volatile storage before restarting the system. The PSHED provides an interface through which the OS can store and retrieve a hardware error record so that the error information is preserved during the system restart. When the system is restarted, the OS retrieves the saved hardware error record so that it can be properly logged in the system event log and user-mode applications can be notified.

Windows also provides hardware error management APIs so that user-mode error management applications can set and retrieve hardware error source information, configure the error handling for a particular error source, and inject simulated hardware errors into the OS for testing purposes.

### What is the impact on Windows Server 2008 R/A/S

The ability to determine the root cause of hardware errors in the past was hindered by the limited amount of error information logged in the Windows system event log. The OS was not capable of preventing system crashes caused by hardware errors, because there was no common error record format and little support for hardware error management applications.

As a result of its error discovery, recording and resolution capabilities, WHEA provides the following benefits:

* Allows for more extensive error data to be made available in a standard format for determining the root cause of hardware errors.
* Provides mechanisms for recovering from hardware errors to avoid halting the system when a hardware error is non-fatal.
* Supports user-mode error management applications and enables advanced computer health monitoring by reporting hardware errors via Event Tracing for Windows (ETW) and by providing an API for error management and control.
* Is extensible, so that as hardware vendors add new and better hardware error reporting mechanisms to their devices, WHEA allows the OS to gracefully accommodate the new mechanisms.

All these benefits assist in a faster time to identification and resolution and the potential for non fatal hardware errors to be identified and resolved without affecting the system’s availability.

## User Account Control

### What is User Account Control

In Windows Server versions prior to Windows Server 2008, when a user with administrative privileges is logged on to the server’s console, the account is operating with its complete privileges active all the time.

This means that if the user attempts to execute a privileged operation (such as installing an application or changing a network configuration) the operation is executed without any warning or prompt about the use of such privileges other than the ones provided by the specific implementation of the application doing the change. Moreover, if a privileged operation is attempted by an application without the knowledge of the operator (such as when a Trojan application is accidentally executed, or a script with unintended results is launched) the operation is performed without indication.

In Windows Server 2008, User Access Control is a technology that prevents unintended execution of privileged operations and provides the capability of performing controlled privileged operations by non privileged users.

### How does it work

When logging locally into Windows Server 2008 as a non administrator (provided that the user has rights to log on locally at the server), a token containing only the minimal privileges is assigned to the session so the user is incapable of making privileged changes. If the user account has administrative privileges however, two separate tokens are assigned. The first token contains all privileges typically awarded to the account, and the second is a restricted token with minimal privileges. All applications are then by default launched with the restricted token, including Windows Explorer, Internet Explorer and the command line. When an application requests elevation or is run as administrator, UAC will prompt the user for confirmation in a Secure Desktop and launch the process using the unrestricted token if the user accepts the request. In the case of non administrative users the prompt also requests administrative credentials to use for executing the operation.

In the Secure Desktop the entire screen is grayed and blocked and only the authorization window presenting the elevation prompt is highlighted.

If the user rejects the elevation prompt (for example if the application being executed wasn’t expected to perform a privileged operation) the elevation is rejected and the application continues without the requested privileges.

### What is the impact on Windows Server 2008 R/A/S

The impact of UAC in Windows R/A/S is twofold.

First, accidental execution of privileged operations (such as disabling a network interface, stopping a service or restarting a server) can produce a considerable reduction in availability. By providing a consistent warning before executing any privileged operation the risk of accidental changes is diminished.

Second, potential execution of malware poses a significant risk for reliability and availability even in servers. For example, when browsing the Internet from a server (a practice that is discouraged by good operation practices but a common one nonetheless) the risk of executing unauthorized code is significant and can result in unintended configuration changes, the installation of Trojan or virus applications or the propagation of viruses. In Windows Server 2008 with UAC enabled, even an account with full administrative privileges (other than the standard “Administrator” account) will run with minimal privileges, and in the case of an attempted privileged operation the UAC prompt will signal the administrator and prevent unintended effects on the operations.

## Service Hardening

### What is Service Hardening

In Windows Server versions prior to Windows Server 2008 services ran mostly under normal user accounts. Since Windows Server 2003 many services run under specialized accounts such as Network Service and Local Service that reduce the potential for abuse of a service privileges but for many applications that install services that must run under a standard privileged user account the risk for privilege abuse, either by code injected in the service (in the case of a vulnerability such as a buffer overrun) or by account hijack is still considerable.

Service hardening is a set of technologies that reduce the privileges of a running service and allow for finer lock down of the resources used by a service.

### How does it work

In Windows Server 2008 when a service is started the set of privileges actually needed by the service account for the service operation must be explicitly informed by the service. The Service Control Manager then computes the intersection between the services assigned to the service account and the privileges required and assigns only those privileges to the account, all other privileges being stripped. This is what is called “privilege stripping” and ensures that a service runs with the minimal privileges necessary minimizing the potential for privilege misuse.

In addition, when a service is configured it is assigned a special Security Identifier independent to the one of the assigned Service Account. This SID is used for the service process to access any resources such as file objects. Thus, it is possible to assign permissions to those resources by configuring Access Control Lists that map to the service instead of the service account. Since the Service SID can only be used by the running service, by ACLing the resources to the service it is guaranteed that only the service will be able to access those resources, and not even other services or application using the service account credentials will be able to read or modify those resources.

Finally, service failure detection and recovery capabilities have been significantly enhanced in Windows Server 2008. If a service fails, the Service Control Manager can perform a failure action, such as restarting the service in an attempt to recover from that failure. In earlier versions of the OS, the definition of service failure was limited to the process crashing in any state other than SERVICE\_STOPPED. With Windows Server 2008, a service doesn't have to crash to have the SCM initiate a failure action. Services can notify the SCM to initiate a failure action if they discover a nonfatal error condition such as a serious memory leak.

### What is the impact on Windows Server 2008 R/A/S

By limiting the privileges used by a service to the minimum necessary it is possible to limit the damage done by a service that’s misbehaving either by design or because of malware attacking the service. Such damage could impact the whole systems availability or might introduce opportunities for further propagation of the malware affecting reliability. A service with lowest privileges offers a much smaller opportunity for such damage.

The ability to use Service SIDs to assign permissions to resources allows for a better lock down of those resources, reducing the threat of other code running in the machine damaging critical service data, and potentially enhancing the services reliability.

The capability for the Service Control Manager to recover services that had nonfatal errors can boost reliability of appropriately designed services that can use this functionality to recover to a working, clean state when they detect anomalous behavior.

# New availability Features in Windows Server 2008

## Kernel Mode Code Signing

### What is Kernel Mode Code Signing

In Windows Server versions prior to Windows Server 2008 it was possible for an administrator to install drivers or kernel components from third parties into the OS with little restriction.

While this allowed for great flexibility it created some serious problems when kernel drivers or components of dubious quality were installed on a system. A failure or vulnerability in the kernel can have significant impact on the whole systems availability and reliability.

Kernel Mode Signing requirement in the 64 bit versions of Windows Server 2008 implies that only binaries signed by a Microsoft approved certification authority can be run at the kernel level.

### How does it work

Kernel Mode Signing can be also performed by software publishers that obtain a Software publisher Certificate from an approved Certification Authority. Approved certification authorities require compliance with certain code quality policies to the software publishers to which the certificates are issued. By preventing unsigned kernel drivers and kernel components from running on a 64 bit server, the quality bar for code running at the kernel level is raised, so chances of low quality or malicious code affecting the whole system are reduced and the chances of rogue malware installing code that runs at this level are significantly reduced. Due to compatibility requirements with older software and drivers running unsigned kernel drivers and components on 32 bit systems is not prevented, though highly discouraged.

As an alternative to obtaining a Kernel Mode Code Signing certificate, a software publisher can submit their drivers to the Windows Hardware Quality Labs for certification. During this certification the drivers are tested for quality and compatibility with a wide range of systems, including different hardware configurations and OS configurations. The Windows Logo Program demands thorough and systematic testing of services, drivers and kernel components before issuing the keys necessary to sign kernel code. These keys also act as code signing certificates, thus allowing the components to be installed in a 64 bit system.

### What is the impact on Windows Server 2008 R/A/S

A significant portion of server failures are due to low quality or poorly tested kernel mode video, network or other drivers. Low quality kernel services such as antivirus, filters or other system level components are also responsible for an important number of server failures. By reducing the chances of low quality code coming from untrusted publishers running at the kernel level the overall system availability can be boosted.

In addition to this most rootkits (self hiding malicious code) and many other malicious applications require running at the kernel level to function appropriately. By preventing unsigned code from running at this level, many such threats are prevented from affecting a system’s reliability.

## Self Healing NTFS

### What is Self Healing NTFS

Disk corruptions can happen for any number of reasons; a few of the more prominent ones are IO errors due to improper shutdown, physical disk surface degradation and software bugs. This type of error can leave the file system in an inconsistent state that might prevent access to the data until some form of corrective action is taken to get it back to a consistent state. The normal process for fixing disk structure errors is running the Check Disk tool (CHKDSK)

Running Chkdsk.exe to fix disk errors is not always completely practical due to a series of issues:

* In many cases CHKDSK will not be able to fix all errors automatically or even identify all the errors while the server is active due to need to access files that are in use. This can be disruptive to the server’s users because chkdsk.exe in repair mode requires exclusive access to the troubled volume.
* CHKDSK can in many instances take a long time to fix all errors.

As disk corruptions are a function of the amount of metadata stored on the volume they will only get more common as volumes grow in size transforming this into a potentially important problem. A self-correcting file system that continues its operations uninterrupted is then an extremely attractive feature for highly reliable and available servers.

The self healing NTFS is a set of enhancements to the NTFS file system code base to support the real-time correction of detected corruption to the on-disk meta-data.

### How does it work

NTFS verifies meta-data as it is read and processed from the disk. Currently when NTFS detects meta-data corruption it raises an error status indicating corruption. With self-healing NTFS at the point a corruption is detected the file system will determine the scope of the repair needed and if the detected corruption is in the list of corruption types it can handle take appropriate corrective action. Such disk correction happens in the background and does not affect the servers operation. If the detected corruption is not of one type that can be handled automatically the volume is marked as dirty and traditional CHKDSK will be required to be run.

Self healing logs an event when a metadata corruption is detected and fixed. The self healing feature can be managed through the FSUTIL utility.

Not all errors that can be corrected by the traditional CHKDSK are covered by the self-healing feature today, but work to add further capabilities to detect and fix more corruption cases is in progress.

In addition to self healing NTFS, traditional stand alone CHKDSK has received several improvements, including the ability to be aborted, better results reporting and logging, performance improvements and more accurate progress reporting.

### What is the impact on Windows Server 2008 R/A/S

For large volumes, execution of CHKDSK can take several minutes and in some rare cases even hours. By performing the checks online and correcting errors without the need for exclusive access to the volume, downtime is averted and whole system reboots are prevented.

## Failover Clustering enhancements

### What are the enhancements in Failover Clustering for Windows Server 2008

Failover clustering in Windows is the ability to pair two or more servers (called cluster nodes) in a group that’s able to perform failover of resources between them either in case of a failure that makes a server unable to host the resource or in case of manual indication such as when patching is being performed on one of the servers or during software upgrades.

Resources that can be failed over are IP addresses, physical disk, network names, services, applications and other such logical resources (in the case of physical disks, the physical units are obviously not moved from one server to the other, but failover clustering assigns ownership of a disk resource to one node at a time, and arbitrates the access to the unit and the mounting of the volumes so only the server actively hosting the resource can see the contents of the drive preventing corruption to the disk due to independent writing to the disk from different nodes).

The resources are normally grouped and linked among them so the complete view of an application or service forms an indivisible unit that moves between servers as necessary.

The clustering services also provide for the ability to detect a server or a resource failure such as the inability to host an IP address (in case of network failure for example) or an application crash. After detecting such a failure the clustering services are able to take corrective measures, whether that’s restarting the resource or application or moving the group to another server in the cluster.

Servers in a Windows Server cluster have to be connected between them in three ways:

* They have to share a network connection that brings access to clients or other servers
* They have to share a network connection that allows them to talk to each other and coordinate operations, including assessing cluster state
* They have to share access to a common disk subsystem through Serial Attached SCSI (SAS), iSCSI, and Fiber Channel (parallel SCSI is no longer supported in Windows Server 2008 clustering)

In previous Windows Server versions these three connections impose specific limitations such as the need for all the servers in a cluster to be in the same subnet, maximum latencies for the links and need for all the nodes to have access the same logical disk units for at least one unit (called the quorum). These restrictions made clusters difficult to deploy in geographically disperse configurations.

In addition to this, clusters were difficult to configure as lots of specific hardware and software configuration requirements had to be strictly met, and chances of a misconfiguration that might later cause operational problems in the cluster were important. In fact, almost 50% of cluster support cases have been in the past related to misconfiguration issues.

Windows Server 2008 addresses both these issues and these enhancements, together with other improvements to clustering functionality, increase the clusters reliability and provide for the implementation of more flexible solutions that increase services availability.

### How do the enhancements in Windows Server 2008 clustering work

In Windows Server 2008 clusters it’s possible to define resources in a way that they have dependencies on alternative resources so that if one resource is not available but an alternative one is, the resource having the dependencies will still function. This allows for the definition of groups (called “Failover Instances”) that contain more than one IP address resource each on a different subnet, enabling each node to be installed on a different LAN with routing between them.

Cluster intercommunication resources can also span different subnets so there’s no need for establishing VLANs that encompass all the nodes. In addition to this, heartbeat timings and tolerances are now configurable, so it is now relatively easy to implement a cluster with nodes located on independent LAN segments and even in networks located in different cities or countries. DHCP assignment of IP addresses is also possible.

High end storage hardware has provided the ability to replicate data in a reliable way across geographically separated storage units for a long time, but this functionality wasn’t enough to cover completely a failover cluster’s needs. A cluster needs to be able to know in a deterministic way which nodes are in operation at any given time, even in the case of a total loss of communication between the nodes. Failure to do so would create risks of more than one node operating independently of the others and performing duties that might be later impossible to reconcile, such as selling the same item or allowing independent and conflicting extractions of money from a bank account. In previous versions of Windows this arbitration was done through quorum resources. One form of quorum resource was an individual disk resource which was subject to arbitration among the active nodes in a cluster. If one node lost communications with the rest it had to attempt to acquire access to the quorum resource (only one node can own a cluster disk at any specific time) and if unsuccessful it would disable itself to prevent conflicts with other live nodes. As this resource has to be guaranteed uniqueness, replicating it among separate storage units presents its own challenges and dangers.

Another type of quorum arbitration was done through majority of nodes. Each node was assigned one “vote” and if a node could not communicate with other nodes that added to a sum of votes of over 50% of the total, it disabled itself. This was only practical in situations where more than two nodes were available.

In Windows Server 2008 clustering arbitration is done through a new quorum model that allows for weighted voting over both disk resources and nodes, plus an additional resource consisting on a remote file share. This last quorum object is ideal for geographically distributed environments where two node clusters have to have the maximum independency possible.

To address the potential misconfiguration issues Windows Server 2008 implements a new installation process. There’s a new test tool that can be run to confirm correct configuration and complete hardware compatibility. This tool can be run during the installation process the setup wizard to insure the cluster not only complies with all the defined requirements, but it also functions correctly according to the cluster needs. This should prevent most configuration issues from affecting production clusters.

These capabilities are complemented by general enhancements in the clustering architecture and code, including better handling of SAN communications and networking capabilities that reduce the number of potential issues and enhance the ability to resolve problems.

### What is the impact on Windows Server 2008 R/A/S

Failover clustering is used to provide high availability to Windows based servers by providing failover capabilities, but it can only help if the cluster is correctly configured and if at least one node survives the failure. The enhancements in quorum arbitration and networking enable easier implementation of geographically disperse clusters which should be a great aid in providing guarantees of node survival in case of power, networking or general building failures, while the new setup and validation process provides greater guarantees of correct cluster configuration significantly reducing the number of cluster failures.

# Serviceability improvements in Windows Server 2008

## Reliability and performance monitoring

### What is reliability and performance monitoring

Windows Reliability and Performance Monitor is a Microsoft Management Console (MMC) snap-in that combines the functionality of previous stand-alone tools including Performance Logs and Alerts, Server Performance Advisor and System Monitor. It provides a graphical interface for customizing performance data collection and Event Trace Sessions.

It also includes Reliability Monitor, an MMC snap-in that tracks changes to the system and compares them to changes in system stability, providing a graphical view of their relationship.

### How does it work

In the Reliability and Performance Monitoring console Data Collector Sets group data collectors into reusable elements for use in different performance monitoring scenarios. Once a group of data collectors are stored as a Data Collector Set, operations such as scheduling can be applied to the entire set through a single property change. Default Data Collector Set templates are provided to help system administrators begin collecting performance data specific to a Server Role or monitoring scenario immediately.

Performance Logs and Alerts features have been incorporated into the Windows Reliability and Performance Monitor for use with any Data Collector Set. Adding counters to log files and scheduling their start, stop, and duration can now be performed through a Wizard interface. In addition, saving this configuration as a template allows system administrators to collect the same log on subsequent computers without repeating the data collector selection and scheduling processes.

The home page of Windows Reliability and Performance Monitor is the new Resource View screen, which provides a real-time graphical overview of CPU, disk, network, and memory usage. By expanding each of these monitored elements, system administrators can identify which processes are using which resources. In previous versions of Windows, this real-time process-specific data was only available in limited form in Task Manager.

The reliability component of the Reliability and Performance Monitoring console, called the Reliability Monitor, calculates a System Stability Index that reflects whether unexpected problems reduced the reliability of the system. A graph of the Stability Index over time quickly identifies dates when problems began to occur. A System Stability Report provides details to help troubleshoot the root cause of reduced reliability. By viewing changes to the system (installation or removal of applications, updates to the OS, or addition or modification of drivers) side by side with failures (application failures, OS crashes, or hardware failures), a strategy for addressing the issues can be developed quickly.

Windows Reliability and Performance Monitor in Windows Server 2008 also provides the same type of diagnosis reports found in the Server Performance Advisor in Windows Server 2003. Report generation time is improved and reports can be created from data collected by using any Data Collector Set.

### What is the impact on Windows Server 2008 R/A/S

Reliability and Performance Monitoring in Windows Server 2008 allows for faster troubleshooting and identification of root causes of intermittent and difficult to isolate problems. By enabling correlation between system changes and reliability or stability issues it allows for fast identification of changes that introduce application or OS problems, helping reduce time to resolution and often reducing the number of incidents that have to happen before a problems cause is identified.

The enhanced data collection and monitoring capabilities allow for better monitoring and diagnosis of performance and stability issues especially when not using specialized monitoring tools such as Microsoft System Center Operations Manager, helping prevent system or application failures and assisting in the diagnosis of problems.

## Windows Eventing 6.0

### What is Windows Eventing 6.0

Since Windows NT 3.1, Windows has incorporated a strong and powerful event logging model that allowed for strong diagnosis and monitoring capabilities. These capabilities were used mostly by administrators for troubleshooting tasks as well as for monitoring security activity and application and system level problems.

Despite its power, the eventing capabilities in Windows versions prior to Windows Server 2008 were limited in scalability and performance and their security granularity. It was also somewhat difficult to use since identifying sequences of events related to an application or service involved the manual application of filters and correlation was not always easy.

Windows Eventing 6.0 implements a new eventing architecture, a new UI for the Event Log and a more modern representation format for the events, solving the performance, scalability and security limitations of the previous model while providing a much more flexible and easy to use interface that simplifies the execution of common tasks and eases the identification of problems and potential issues.

### How does it work

The new Windows Eventing infrastructure implements a new architecture for event processing. Now event recording is asynchronous allowing for high volume events being recorded without affecting the performance of the recording application or service, while implementing immediate processing and publishing for lower volume event types. Events can be now delivered in addition to the event log to event subscribers in real time. Two such event subscribers are the Task Scheduler, which can now perform tasks in response to an event, and the Event Forwarder, used to send events to a remote event collector.

The new events have a well-defined and published structure, allowing for consistent filtering and searching of events. The events now include consistent attributes including holders for severity, source sub component, activity opcode and keywords, among others.

The events are now presented externally in XML with a published schema. A new query language based on XML Path Language expressions (XPath, see <http://www.w3.org/TR/XPATH>) for more information) is available for obtaining filtered lists of events. This query language can be used in conjunction with the rich set of standard attributes in the events to create queries that present the desired information in a simple and compact way. These queries can be used for extracting information from the event logs, for associating tasks to events, for forwarding select events to a specific remote event collector or for various other potential uses.

The new UI for the Event Viewer now provides more organized information. In addition to enhancements in the interface, Event Viewer is now implemented in a MMC 3.0 console with preview and simultaneous multiple event review capabilities, and there are new log views available. In particular, there are now separate event views for services and applications running in the system, allowing for easily identifying and isolating events related to a software component, and there’s the capability for defining custom stored views including events matching a specific query.

### What is the impact on Windows Server 2008 R/A/S

Event logging is a critical tool for managing a system, reducing outages and accelerating the resolution of problems.

Besides bringing enhanced scalability and performance to the event logging process, the new capabilities in Windows Eventing 6.0 have the potential to reduce the number and duration of outages to a system.

The new UI allows for easier isolation of problems and better identification of root causes, increasing availability. The ability to define custom views and per application views provides better problem detection capabilities. The ability to associate event filters to tasks in the task scheduler allows for some out of the box reaction capabilities to events, enabling automatic resolution of some types of problems and positively affecting availability. The new user interface allows for easier servicing of Windows Servers by enabling a better understanding of a servers status and behavior.

## Hot Patching

### What is Hot Patching

When an OS update is installed on a Windows Server the update normally modifies code and data in the machine. If the updated component is active at the time of the update it is likely that data files will be in use and that the code will be loaded in memory and running. In the past this implied that for OS component updates the machine often had to be reboot to make the changes active, as open files would not be replaced until the server rebooted and the code running in memory would not be updated without restarting the services. While it was not always necessary to reboot the machine immediately to finalize a patch installation, the new behavior would not be in place until the reboot was performed and the affected files were updated during the shutdown or startup phase, leaving the system in danger especially in the case of critical security updates.

Hot patching is a technology that allows for the application and immediate activation of certain kinds of updates without the need for a reboot.

### How does it work

Hot patching technology provides dynamic servicing of system files without requiring reboots. The technology is available under Windows Server 2003 SP1 and is now integrated in the OS and significantly enhanced in Windows Server 2008.

Under Windows Server 2008 the unit of servicing is a component. A component is a set consisting of files and registry settings. A component’s manifest completely describes the composition of a component. An update contains its own manifest that includes the information on the components to be updated and the update pieces. The update manifest is used in conjunction to the component manifest to decide whether the installed component can be hot patched in a particular system and the steps to be followed to perform the operation.

With hot patching if an active component is deemed to be hot patchable the system performs a replace of the corresponding files, even if the files are currently in use, and then injects the fix in the affected running modules as loaded in memory. The running component is updated and adopts the corrected behavior without the need for a reboot.

The update process is completely transactional, so even in the case of a failure in the process of applying the update to the file system the system will roll back the changes without interruption. If an error in the patching of the in memory components occurs, the components in memory might not be fully patched until the next reboot, when the cold patched files will be loaded normally.

### What is the impact on Windows Server 2008 R/A/S

One of the most frequent reasons for server reboots is the application of software updates. Hot patching can reduce this need significantly. It is estimated that, of the patches released for Windows Server 2003 that required a reboot, between 30% and 50% (depending on the period analyzed) would have been able to be hot patched. Since, according to certain studies (see Appendix B) almost 50% of server reboots on modern Windows servers are due to software updates, this feature should allow for a considerable reduction in the number of server reboots of Windows Server 2008 based servers.

## Server Core

### What is Server Core

Server Core is a new installation option in Standard, Enterprise, Datacenter and Web editions of Windows Server 2008 that allows for a minimal installation of the OS with only the necessary components to run specialized workloads.

### How does it work

When installing Windows Server 2008 the option is given to install a full OS or just a minimal “Server Core” installation.

If the Server Core installation option is selected, only the OS kernel, minimal user interface elements and minimal services are installed and configured. The remaining OS components and services are not installed and not copied to the server.

By installing a minimal set of components in a server the attack surface is reduced and the number of components that need configuration, maintenance and updating is minimized. Most attacks on servers, aside from application level attacks, are performed by exploiting either vulnerabilities in code or misconfigurations. Reducing the number of components installed helps significantly reduce potential problems.

Components not installed on a system do not need to be updated. Reducing the number of components installed reduces in turn the number of patches that need to be applied, and the number of reboots needed because of these patches.

Having only minimal components installed a Server Core installation is not apt to be used in all workloads. At the moment only Domain Controllers, DNS, DHCP, File Server, AD LDS, Media Services, Print services and Windows Virtualization Services roles are available on a Server Core installation. WINS, Failover Clustering, Subsystem for UNIX-based applications, Backup, Multipath IO, Removable Storage Management, Bitlocker Drive Encryption, SNMP, and Telnet Client are also supported in a Server Core installation. Internet Information Services will also be supported in a restricted configuration without ASP.NET support.

In the absence of a Graphic User Interface (GUI), Internet Explorer or Windows Explorer shell, and local MMC support is, most management operations have to be performed either remotely or through the command line applications.

### What is the impact on Windows Server 2008 R/A/S

By reducing the number of components installed in a Server Core installation, the need for configuration and patching is significantly reduced. Based on patching metrics from previous versions of the OS, about 60% of bugs patched were in components not included in a Server Core configuration, indicating that a high reduction in attack surface can be expected in Server Core installations.

## Internet Information Services 7

### What are the Reliability, Availability and Serviceability enhancements in Internet Information Services

IIS 7.0 includes significant changes in the web server architecture, providing a more robust, scalable and reliable platform for servicing web applications.

In particular, IIS 7 includes the following enhancements that affect Reliability, Availability and Serviceability:

* Powerful diagnostic and troubleshooting tools
* Modular request processing pipeline
* Better role customization
* XCopy application and configuration deployment

### How do these changes work

IIS 7 enables developers and IT Professionals to more easily troubleshoot errant Web sites and applications. IIS 7.0 provides a clear view of internal diagnostic information about IIS, and collects and surfaces detailed diagnostic events to aid troubleshooting problematic servers.

The changes in IIS 7.0 allow a developer or an administrator to see, in real time, requests that are running on the server. Now, it is possible to filter for error conditions that are difficult to reproduce and automatically trap the error with a detailed trace log.

IIS 7.0 includes a new Runtime State and Control API, which provides real-time state information about application pools, worker processes, sites, application domains, and even running requests. This information is exposed to COM, WMI, command line tools and the management console.

IIS 7.0 also includes detailed trace events throughout the request and response path, allowing developers to trace a request as it makes its way to IIS, through the IIS request processing pipeline, into any existing page level code, and back out to the response.

For application deployment and servicing, IIS 7.0 allows for storing all of the IIS configuration settings in web.config files, which makes it much easier to deploy applications by just copying the folders containing the data, code and configuration files. This process is called XCopy deployment. Instead of having to run scripts or manually configure settings appropriate for each application or component, in XCopy deployment a whole application is copied to the destination folder and the new configuration, contained in the web.config files inside the folder, become active. It is possible to use XCopy deployment to deploy applications across multiple front-end Web servers, thereby avoiding costly and error-prone replication and manual synchronization issues.

The IIS 7.0 core Web server (not to be confused with Server Core, discussed also in this document) includes some fundamental changes from IIS 6.0. For example, both native and managed code is processed through a single request pipeline. In addition, IIS 7.0 features a Web server engine that permits addition or removal of components, called modules, depending on their need.

In previous versions of IIS, all functionality was built-in by default, and there was no easy way to extend or replace any of that functionality. The IIS 7.0 core is instead divided into over 40 separate feature modules, and additional modules can be written and added to IIS. Because all IIS core server features were developed as discrete feature modules, it is possible to add, remove, or even replace IIS feature modules.

### What is the impact on Windows Server 2008 R/A/S

The new diagnostic and troubleshooting tools should allow for easier identification and faster resolution of application problems, providing the ability to reduce time to solution. The detailed trace events allow developers to understand, not only the request path and any error information that was raised as a result of the request, but also elapsed time and other debugging information to assist in troubleshooting all types of errors, and when a system stops responding.

XCopy deployment not only reduces the effort necessary to deploy and maintain applications, but should also reduce errors in deployment by allowing for simpler automation of deployment- especially in environments with large number of web servers and several application layers. Also, by keeping application code together with application configuration, it is less likely that mismatches between code and configuration will occur when moving an application through the different stages in its lifecycle.

The new modular Core Web Server architecture allows for more customized installations, reducing attack surface and maintenance needs, including need for patching, component hardening and configuration. As a result, an appropriately customized web server should offer more resiliency to attacks and have lower administration needs.

## Hot Addition and Replacement of CPU, Memory, ACPI buses and cards

### What is Hot Addition and Replacement of CPU, Memory, ACPI buses and Cards

When a service that cannot scale horizontally by adding more servers starts approaching the capabilities of the hosting machine, the option of adding additional CPUs, memory or I/O interfaces is often the most efficient one.

This operation often means shutting down the machine, opening it, adding the new resources, powering the machine up and configuring the new resources in the BIOS and the OS.

With Windows Server 2003 the option to hot add memory was included in the OS, allowing for the addition, on servers supporting the capability, of additional memory without interrupting the operation of the server. Hot add of I/O cards was also present in the product for certain types of I/O buses.

In Windows Server 2008 the capabilities for hot adding processors and I/O buses are added.

In addition, for cases when a component needs to be replaced, the capabilities for performing Hot Replace of memory and processors are provided. A typical such need arises when live diagnostics indicate that the components are showing degraded performance or predict imminent failure, allowing for the resolution of such problems before catastrophic failure and without service interruption.

With native support for PCI Express, Windows can support the hot addition and removal of certain types of PCI Express cards. PCI Express is the latest standard bus used in servers based in the x86 and x64 based servers. PCI Express provides high performance and scalability, and several capabilities that allow for greater reliability and serviceability. It is a high speed bus standard using a serial architecture and commonly used in servers as a system bus and I/O interconnect replacing the functions previously provided by PCI and AGP buses.

### How does it work

Hot addition of processors, memory and IO bridges can be provided in servers supporting the capability without halting the systems. The capability must be supported by the hardware by providing the appropriate electrical signals to power down the appropriate slots and detect the insertion of new resources in the slots, among other things.

Once a new physical resource, such as a CPU, a memory unit or an I/O bridge, is connected to a system, Windows will automatically detect the resource and allow for its utilization. Some applications with hard allocation of resources such as processors or memory might have to be enhanced to take advantage of this functionality without service interruption.

Hot replacement of resources allows for the replacement of a resource currently in use. This capability needs both the new resource and the original resource to be in the system at the same time to allow for a seamless transfer of the current load to the new units. This implies that, at least, one additional free slot or an unused unit of the type to be replaced (memory or CPU) must be available in the machine when the replacement is about to be performed.

Windows Server 2008 also provides advanced diagnostic capabilities of defective memory and processors that allow for advanced detection of certain types of component failure which, combined with the hot replace capability, can prevent serious system failures.

Systems capable of creating one or more hardware partitions comprising a subset of its CPUs, a group of memory units and a set of I/O buses are referred to as dynamically partitioned servers. The capability to support hot addition and performing rebalancing of memory, buses and CPUs helps in the adequate allocation of resources to the different partition and allows for dynamic system sizing. Windows Server 2008 incorporates other technologies and enhancements that make the operation on dynamic partitionable servers more seamless and efficient.

Support for hot removal of processor, memory and bus resources is not included in Windows Server 2008.

Thanks to native PCI Express support, PCI Express cards can be added or disabled and removed without halting the OS or shutting down the server. Combined with the Windows Hardware Error Architecture and PCI Express error signaling and error checking and correction capabilities, this provides the ability to detect certain types of failures on some types of components and replace the affected components without service interruption.

### What is the impact on Windows Server 2008 R/A/S

Hot addition of resources to servers allow for scaling up of physical servers without service interruption helping enhance system availability.

Hot replacement of components together with early detection of component degradation can prevent system failure and allow for the replacement of the affected components without affecting uptime and improving reliability.

The ability to hot add and hot remove PCI Express cards provides with better availability and serviceability of the platform, reducing the need for shutdowns for hardware maintenance.

# Virtualization Technologies impact on R/A/S

Server virtualization technologies are being used with increased frequency in production environments. Often quoted reasons for this trend are the improvement in hardware utilization provided by server Virtualization and the greater flexibility and agility these provide in implementing new systems. But there are other significant benefits in implementing virtualization technologies in the areas of availability and serviceability.

Hyper-V, Microsoft’s Windows Server 2008 based virtualization platform, provides several significant capabilities that allow for enhanced availability and serviceability.

All virtualization platforms add at the very least one additional element that can fail to the platform: the virtualization engine. Hyper-V adopts a new Hypervisor type microkernel architecture, which means the guest OSs run on top of a very slim virtualization layer that runs on top of virtualization enabled hardware. Hyper-V still requires a full operating system in the Primary Partition for management and hardware driver operation. The OS in the primary partition can be a full server installation or a Server Core configuration. Most critical components of the virtualization operation have moved to the hypervisor layer, which is extremely slim and contains only the most critical components. This architecture reduces the potential for platform failures and vulnerabilities compared to virtualization solutions built on top of complete OSs, increasing the potential reliability and availability of the virtualized workloads.

In addition to the architectural enhancements, Hyper-V provides support for clustering technologies both at the host and the guest level that enable new levels of application availability.

By supporting guest clustering within a single or between multiple virtualization hosts, whole clusters can be implemented inside the virtualization environment, thus providing the application failure detection capabilities normally only found on physical clusters. This brings the possibility of implementing clustering technologies for applications that on their own wouldn’t justify the investment in a clustering platform, bringing availability of low end applications to new levels.

Host level clustering support is complemented by other technologies to enable live migration of virtual machines from one virtualization host to another. This means that virtual machines can be moved without significant service interruption from one physical system to another, either for load balancing reasons, or to permit hardware maintenance at the virtualization host. Quick migration is performed on a virtual machine by first saving the memory state for the virtual guest to the shared storage and then performing the transfer in a very short period that is not normally noticed by users. This capability is supported by Clustering Services at the host level to allow for seamless access to virtual disk resources on the guest. Quick migration can bring failover support to applications and legacy OSs that do not support failover capabilities natively.

A virtualization environment such as Hyper-V also provides benefits for serviceability and manageability. Backup operations that are normally performed inside an OS can now be performed at the host level. With Hyper-V a single backup operation performed at the host level can back up, through synchronized Volume Shadow Copy operations at the host and guest level, the host OS, data files, guest OSs and application level data such as Active Directory and SQL Server databases in a transactionally consistent, point in time view. This capability not only simplifies the back up operations of the virtual environment but also reduces the chances of data being accidentally excluded from a back up or backups of different related systems being taken at different points in time, thus leading to difficulties reconciling restored data.

# Other R/A/S enhancements

In addition to the code quality improvements, features and enhancements described above, there are many other changes that can improve R/A/S of Windows servers that are included in this release. Some of these changes are described below.

One thing that can seriously reduce a server’s availability and reliability is malicious hacking. One technique often used by attackers to obtain control of a server is to execute code already existing on the system in an unintended way, such as the intentional overwrite of a pointer through an unchecked buffer so it points to code in a known position. Another related technique consists of modifying a pointer to a buffer so it overwrites existing code in memory that’s known to be executed in the future, such as the code for a system call. Both techniques commonly rely on knowing the addresses at which system DLLs are loaded.

Prior to Windows Server 2008 Kernel, HAL, executables, and DLLs loaded at fixed locations. In Windows Server 2008 modules are based at one of 256 random points in the address space so the chances for a process to know where a piece of code is exactly located are greatly reduced. This technology applies to kernel components, the hardware abstraction layer and even kernel level drivers. User stack locations are also randomized, greatly reducing the effectiveness of such attacks.

Specific enhancements in the kernel and other components were made to support systems that include error handling capabilities. As an example of this, Fault Tolerant servers have been available for some time, but providers of such hardware based their implementation on special drivers or software. The fault tolerance solutions included little or no direct support by the OS. Windows Server 2008 incorporates features aimed to supporting Fault Tolerant servers, including differential memory synchronization, which allows for the memory and state of two identical systems to be synchronized in order to allow for instantaneous transfer of load in case of a hardware failure on the active system.

Some systems are not designed to be fault tolerant, but still include components that provide hardware diagnosis capabilities that can help avert downtime and data loss. One such component is S.M.A.R.T. capability included in most new hard drives.

Windows Server 2008 can potentially eliminate much of the impact of a disk failure by detecting disk problems in advance, before total failure occurs. Hard disks often show warning signs before failure and report them using the S.M.A.R.T. capability, but earlier Windows OSs did not act on these warning signs. Windows Server 2008 listens for evidence that a hard disk is beginning to fail and records the events in the event log for analysis by monitoring tools.

Another technology now included in the OS related to redundant hardware is Multipath I/O. Multipath I/O (MPIO) is a technology used in storage area networks that allows multiple communication paths between servers and storage devices to be assigned for redundancy and load balancing. Multipath I/O technologies have been available for a while in the form of proprietary solutions which often were difficult to set up and configure. Many proprietary Multipath I/O solutions depended acquisition of specialized hardware and software, often limiting interoperability between multiple vendors’ devices. Since a completely generic OS level solution could have been limited in the ability to exploit specific hardware features, Microsoft opted for Windows Server 2008 to use a combined approach. The Multipath I/O capabilities are now included in the OS. The MPIO technology works in conjunction with device-specific modules, or DSMs, provided by the hardware manufacturers that interface between the OS Multipath capabilities and the hardware device. The net result is a complete solution that makes use of advanced hardware capabilities without requiring specialized management or implementation.

Nevertheless, even in redundant systems an I/O request can fail or timeout. When a request for I/O fails or does not complete in a specific time, the application that performed the request might still be in a state that enables it to handle the error and continue operating. With the previous Windows driver architecture, it was very difficult for applications to handle the termination of an I/O request and the resumption of operations, as it was uncommon for drivers to provide the specific functionality to allow request cancellation. Because this type of failure to successfully cancel an I/O request from an application can frequently leave the I/O subsystem in a state that requires a system reboot, it is a situation that can severely affect a system’s availability.

To address this problem Windows Server 2008 now supports driver IO Cancellation in two fronts. First, it promotes the adoption of I/O Completion/Cancellation guidelines for drivers that operate on hardware or on remote, loosely coupled devices. These guidelines specify a set of behaviors that drivers should support in order to allow for appropriate and orderly cancellation of I/O requests if they do not complete within a reasonable period. Second, Windows Server 2008 now supports Application-Initiated I/O Cancellation, which is a feature that allows for applications to direct drivers to cleanly cancel I/O requests while leaving the driver and hardware in a working state. Together, these two capabilities, if implemented properly by applications and drivers, can significantly reduce the number of I/O related application hangs requiring a reboot.

Sometimes a bug in a driver or other kernel level component might make it overwrite some random data, which could lead to accidentally changing registry settings in a way that could be difficult and time-consuming to diagnose and correct. Corruption of the registry tends to have a disproportionate impact on overall reliability because this corruption can persist across system restarts.

Windows Server 2008 prevents poorly written drivers from corrupting the registry accidentally by setting the corresponding memory pages as Read Only. When an application has to perform changes in these settings it must do so by following the appropriate channels, which include the necessary temporary change in the page protection status prior to modification. This protection enables the memory management component to achieve protection the vast majority of the time, with low overhead.

In order to increase the quality of drivers available for Windows Server, there have been several improvements to the Windows Logo hardware certification program. These improvements include stricter and more automated tests, and additional driver and hardware requirements. The increased testing now includes stress on large/complex hardware configuration, driver behavior in terminal server environments, bad parameter calls to the drivers and extremely low memory testing among others.

Additional requirements that the drivers must comply with in order to be certified include:

* correctly handling improper requests
* not bypassing system service,
* clean install/uninstall and Load/Unload
* implementation of I/O Completion and I/O Cancellation guidelines
* native x64 support
* making use of the PnP APIs
* supporting Resource Rebalance requests
* Query Power and Set Power Management requests

As for the actual server systems they are now expected to comply with expanded guidelines, including using ECC or other self correcting memory, providing remote, headless and out of band management capabilities, BIOS support for USB peripherals and PCI Express support.

Additional server qualifications have been created for Fault Tolerant servers, Highly Available servers and Dynamically Partitionable servers. Each category of server comes with specific requirements and tests, enabling a more consistent and fine grained criteria for these classes of servers.

Driver development tools have been improved as well, and the Windows Driver development Kit now includes tools such as the Static Drive Verifier tool, which systematically checks all paths in code for incorrect behavior, and PreFast for Drivers, an enhanced code-checking tool optimized for discovering common errors in kernel mode drivers.

Windows Server 2008 also protects system settings from corruption or inadvertent changes that can cause the system to run incorrectly or not run at all. Windows Resource Protection (WRP) protects critical system settings, files, and folders from changes by any source except a trusted installer. This prevents users from changing critical system settings that can render systems inoperable. It also prevents software installers from inadvertently overwriting a system file that has been updated by Microsoft with an older version that came on the distribution media.

Even this type of protection cannot possibly prevent the intentional modification of system files by kernel running code. A new feature that ensures the integrity of the boot process for your servers addresses this scenario. Windows Server 2008 creates a validation key based on the kernel file in use, a specific hardware abstraction layer for your system and drivers that start at boot time. If these files change after this key is created, at any subsequent boot, the OS will know and halt the boot process so execution of unauthorized low level kernel code is averted and the problem can be repaired.

Another type of common memory handling error is that of memory leaks. Memory leaks occur when an application or service is in a loop that makes a memory allocation that is never freed. If the loop goes on for a long time, the drained memory accumulates producing memory exhaustion in the server. Such errors are a significant cause of unresponsiveness, performance problems and system halts due to resource exhaustion.

When the memory manager in Windows Server 2008 detects a series of memory allocations that continuously drain memory from the heap in an abnormal way it logs specific events that can be identified and acted upon by monitoring tools such as Microsoft System Center Operations Manager. A similar capability allows for detection of virtual space exhaustion for an application, allowing for preventive action to be taken.

A different problem that often causes reliability problems is a deadlock, which occurs when two processes or threads are waiting at the same time for an event, but each event depends on the other process to finish to be activated. For example, process A might be waiting for an output from process B, while process B needs the result from process A to complete. Such a situation is not uncommon and it is one of the main causes of applications hanging or freezing. While this situation is most common in interactive applications such as desktop tools, server applications can also suffer from deadlocks and their impact is significant.

Wait-Chain Traversal (WCT) with deadlock detection is a new capability for the OS in detecting application deadlocks and logging the appropriate events, allowing for appropriate action to be taken.

When used in conjunction with a monitoring tool that can take action upon the logging of these situations, preventive measures can often be taken preventing performance and reliability issues. Since in most cases the corrective action involves shutting down the offending process, an ideal combination is combining these features with a system such as Server Clustering. This way instead of continuously degrading the performance of a system or keeping an application in a blocked state the system is able to quickly restart the offending application with minimal impact to users.

Resource exhaustion is also sometimes caused by the fixed size of the Virtual Address space normally assigned to the kernel structures in prior versions of Windows. In Windows Server 2008 the virtual address space for kernel components is dynamically allocated as needed. The Virtual Address Space partitioning is done on the fly with automatic tuning, so no manual configurations reboots are needed for its adjustment, which reduces the likelihood of virtual address space exhaustion.

The server startup and shutdown processes have also been enhanced for increased reliability and serviceability.

For some workloads it is common that services work with large amounts of data which they process using large amounts of memory. When these services stop they often need a long time to write that data to the disk. In previous versions of Windows, services had no way to extend the time allowed for shutdown. After a fixed timeout (default of 20 seconds), even if services were still running, the Service Control Manager was killed and system halted. This was a problem for services that needed to flush data to disk. In Windows Server 2008, services can request pre-shutdown notification. Once they make such request the services can take as long as they need to flush data cleanly and shut down as long as they are responsive. If a service stops responding, the system shuts it down after three minutes. After services that requested pre-shutdown notification stop, the system performs normal shutdown for other services

During the early part of the boot process, prior to Windows Server 2008 system crashes would not result in a crash dump. Crash dumps are written to the paging file on the boot partition, and the paging files were not opened by SMSS process (after kernel initialization), so there was nowhere to write the dump to. In Windows Server 2008, the paging file is open before system start drivers initialize, thus allowing for a crash dump to be generated even if the problem occurs in the early parts of the process. This allows for enhanced troubleshooting and diagnosis of boot problems caused by hardware and mass storage driver errors, in addition to other types of problems that were previously difficult to diagnose. Additionally, Windows now supports crashing to an alternate device connected to any miniport driver, giving administrators more flexibility in performing crash diagnostics.

There are many other changes and enhancements in the OS that provide better reliability, availability and serviceability, and are not covered here. Features such as Storport Hardened for SANs, IO Timeout modified for busy storage targets, long distance LUN Queue Handling, full support of MSI-X enabled host bus adapters, enhancements to Power Management Handling, memory corruption detection through Page Zero checking, the Windows Memory Diagnostics tool, Hardware Watchdog Timers and others are outside of the scope of this document, but all provide an increased capability for Windows Server 2008 to operate in critical environments.

Windows Server 2008 is expected to take Windows to more critical roles than ever. The features and enhancements identified in this paper will increase its capability to support such workloads with minimal downtime, maximum reliability and a level of serviceability never seen before.

# Definition of Failure modes and Windows Server 2008 Enhancement Impact

The following table lists many common causes for system failures that are prevented, averted or alleviated by some of the new RAS features in Windows Server 2008 as described.

A mark for a cell should be interpreted as “this function or capability helps reduce the incidence or impact of this type of incident”.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Kernel Patch Protection | Transactional File System | PCI Express support | User Account Control | Service Hardening | Kernel Mode Code Signing | Windows Hardware Error Architecture | Self Healing NTFS | Failover Clustering enhancements | Reliability and performance monitoring | New event log model | Server core | Hot add and Replace CPU, Memory and Buses |
| Hardware Failures |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Disk or array failures |  | 🗹 | 🗹 |  |  |  | 🗹 | 🗹 | 🗹 | 🗹 | 🗹 |  |  |
| CPU failures |  | 🗹 |  |  |  |  | 🗹 |  |  | 🗹 |  |  | 🗹 |
| Motherboard or bus failures |  | 🗹 | 🗹 |  |  |  | 🗹 |  |  | 🗹 | 🗹 |  | 🗹 |
| Memory failures |  | 🗹 |  |  |  |  | 🗹 |  |  | 🗹 | 🗹 |  | 🗹 |
| Network Interface Failures |  | 🗹 | 🗹 |  |  |  | 🗹 |  |  | 🗹 |  |  |  |
| Software Failures |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Operating System Error | 🗹 | 🗹 |  |  | 🗹 | 🗹 |  | 🗹 |  | 🗹 | 🗹 | 🗹 |  |
| Resource Exhaustion |  |  | 🗹 |  |  |  |  |  |  | 🗹 | 🗹 |  |  |
| Application Error |  | 🗹 |  |  |  |  |  |  |  | 🗹 | 🗹 |  |  |
| Data Corruption | 🗹 | 🗹 |  |  |  |  |  | 🗹 |  |  | 🗹 |  |  |
| Operation Errors |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Configuration Errors |  |  |  | 🗹 |  |  | 🗹 |  | 🗹 | 🗹 | 🗹 | 🗹 |  |
| Installation Errors | 🗹 | 🗹 |  | 🗹 |  | 🗹 |  |  | 🗹 | 🗹 | 🗹 |  |  |
| Handling Errors |  |  |  | 🗹 |  |  | 🗹 |  | 🗹 |  | 🗹 |  |  |
| Intentional Damage (Hacking, abuse) | 🗹 |  |  | 🗹 | 🗹 | 🗹 | 🗹 |  |  |  |  | 🗹 |  |
| External Failures |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Energy Failure |  | 🗹 |  |  |  |  |  |  | 🗹 |  |  |  |  |
| Total Building Failure |  | 🗹 |  |  |  |  |  |  | 🗹 |  |  |  |  |
| Communication Failure |  | 🗹 |  |  |  |  |  |  | 🗹 |  |  |  |  |
| Planned downtime |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Software updates |  |  |  |  |  |  |  |  |  | 🗹 |  | 🗹 |  |
| Hardware Changes or repairs |  |  | 🗹 |  |  |  | 🗹 |  |  | 🗹 |  |  | 🗹 |
| Offline configurations |  |  | 🗹 |  |  |  |  |  |  |  |  |  |  |
| Operation related reboots |  |  | 🗹 |  |  |  |  | 🗹 |  |  |  |  |  |

# Appendix A: System AVAILABILITY Calculation

## Definitions

*Availability:* Probability that a System is Operational during any specific time in its operation period

It is accepted that after a sufficiently long period of time any system will eventually fail. When measured over a long period of time Availability should tend to a fixed numerical value considering it was adequately maintained and repaired with components of identical characteristics.

*Reliability:* Probability that a specific result given that a System is correct and that it was delivered during the expected time slot.

*Operational State*: Measured as Functional or non functional. It can also be measured as a fractional number for a degraded state.

Burn-In and Wear Out: There are many potential failures of a component that will be evident right at first power up or after a short period of time. This is what causes the period known as burn in, during which the probability of failure is very high. This is what is usually known as infant mortality. It is due to this that it is often recommended that every system to be put in production passes first through a burn in period during which most such failures will be detected and resolved.

Other types of failures are related to wear, and this is especially observable in devices including mechanical components (spinning disks, fans, etc.) as well as specific electronic components such as NAND based flash memory. The failure rates of such components tends to go up with time as components wear out, but normally this only happens after a certain period of time (mainstream life of the component) during which the device has an almost constant failure rate.



## Examples

For a hard drive with a mean time to failure (MTTF) of nine months, average reliability for a month of use is:

 R=1-(1 m./9 m.) ~ 89%

(this is an approximation to )

But this is only valid during mainstream life of the component. As the drive becomes older and accumulates wear, the reliability will go down, to the point in which after a certain age failure will become very likely.

Assuming that an operator can detect, diagnose and repair the failure (replace the component) on average on 4 hours (mean time to repair) we can calculate the drive availability as follows:

The process of the disk failing and being repaired can be graphed as:



With two operational states: up and down.

The unit goes from up to down with a frequency of 1/MTBF (during its up time) and from down to up with a frequency of 1/MTTR (during its down period). For each cycle up/down which lasts on average MTBF+MTTR, the system is in the “up” state during MTBF time, so it results that A = MTBF/(MTBF+MTTR).

 A = MTTF/(MTTF+MTTR) = 9 months/(9 months+ 4 hours)

 = 99.4%

For more complex systems, such as a disk array composed of two disks in RAID 1 configuration, we can represent the status as:



Given that the system will be operational with only one of the disks working we can consider “operational state” the “2 up” and “1 up, 1 down”. Availability is thus:

A = Prob(2 up)+Prob(1up, 1 down) = 

The “degraded state” probability can also be calculated in a similar way.

For a complex system we can compose availabilities to get to the final availability. For a simplified case:

A SQL Server with a desired availability of 99.9% implies:

Availability= Prob(SQL Platform up) = Prob(SQL App. up/ NT up) \* Prob(NT up/ Server HW up) \* Prob(Server HW up) = 99.9%.

Prob(X/Y) should be read as “Conditional probability of occurrence of X given the occurrence of Y”.

For a single server we can consider:

End user availability <= Application availability <=OS availability <= Hardware availability

Any system is thus a series of components working in parallel and in series. We can model such systems (assuming failures of components A and B are independent) as follows.

Components in Series:



(for example, A=controller and B= disk drive)

Components in Parallel:

A

B

A

B



(for example, A and B are two disks in a RAID 1 or two servers in a Network Load Balancing group).

It is important to consider that individual MTBF numbers can only be applied for units not suffering from cumulative wear. In devices (or systems containing devices) that have mechanical or electrical wear (spinning disks, NAND flash memory, fans, etc.) such calculation is only valid for new or young units. For older units, the likelihood of failure, as expressed in the previous section, goes up with time. And this is especially important with units installed at the same time: a simultaneous failure of two identical equally old units it is much more likely than what calculation based on simple failure rates would indicate.