

# Memory Sizing Guidance for Windows 7

July 14, 2009

## **Abstract**

---

This paper describes an approach that information technology (IT) professionals and system builders can use to determine how much memory is required to successfully run a set of programs. It explores how the choices in machine architecture, hardware devices, driver version, and memory configuration affect the amount of memory that is available to users' programs.

By following the guidelines in this paper, you can determine the appropriate amount of memory to configure for anticipated customer workloads.

This information applies to the Windows® 7 operating system.

References and resources discussed here are listed at the end of this paper.

The current version of this paper is maintained on the Web at:

<http://www.microsoft.com/whdc/system/hwdesign/MemSizingWin7.aspx>

**Disclaimer:** *This is a preliminary document and may be changed substantially prior to final commercial release of the software described herein.*

*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.*

*Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, email address, logo, person, place or event is intended or should be inferred.*

© 2009 Microsoft Corporation. All rights reserved.

*Microsoft, MSDN, SuperFetch, Windows, and Zune 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.*

#### **Document History**

<b>Date</b>	<b>Change</b>
July 14, 2009	Final Draft

## Contents

---

Introduction .....	4
Memory Usage .....	4
Displaying Memory Usage by Using Resource Monitor .....	5
Commit Charge .....	6
Process Working Set .....	6
Pages Not Included in a Process Working Set .....	6
Trimming a Process's Working Set .....	7
Page Faults.....	7
Global Memory Usage .....	8
System Reference Set .....	9
Results of Memory Use Analysis on Sample Systems .....	10
Determining the Required Memory for a Configuration .....	17
Methodology Background .....	17
Testing the Configuration .....	17
Step 1: Configure the Hardware.....	17
Step 2: Configure the Software .....	18
Step 3: Create a Test Script .....	18
Step 4: Configure Data Capture Tools .....	18
Step 5: Run the Test .....	19
Analyzing the Results .....	19
Best Practices for Memory Performance Testing .....	20
Resources .....	20
Appendix A. Tested Configurations.....	22
RAM: Installed versus Recognized .....	22
UMA Graphics Hardware .....	23
Appendix B. Tools for Measuring Memory Usage.....	24

---

## Introduction

---

This paper provides an approach that information technology (IT) professionals and system builders can use to determine how much memory is required to successfully run a set of programs. It explores how the choices in machine architecture, hardware devices, driver version, and memory configuration affect the amount of memory that is available to users' programs. The paper covers:

- How Windows® 7 categorizes memory usage and what each category means.
- How to determine the size of the different memory categories and how to measure the amount of memory that is available to applications.
- Factors that affect the amount of memory that is available to applications.

Original equipment manufacturers (OEMs), information technology (IT) professionals, and enthusiasts can use this information to understand how Windows uses memory and determine the optimal memory size when they configure a PC to run Windows 7.

Readers should be familiar with the basics of the Windows memory management as described in *Windows Internals*, which is listed in "Resources."

**Note:** The analysis in this paper is based on a prerelease version of Windows 7. Results for the final version of Windows 7 may vary.

---

## Memory Usage

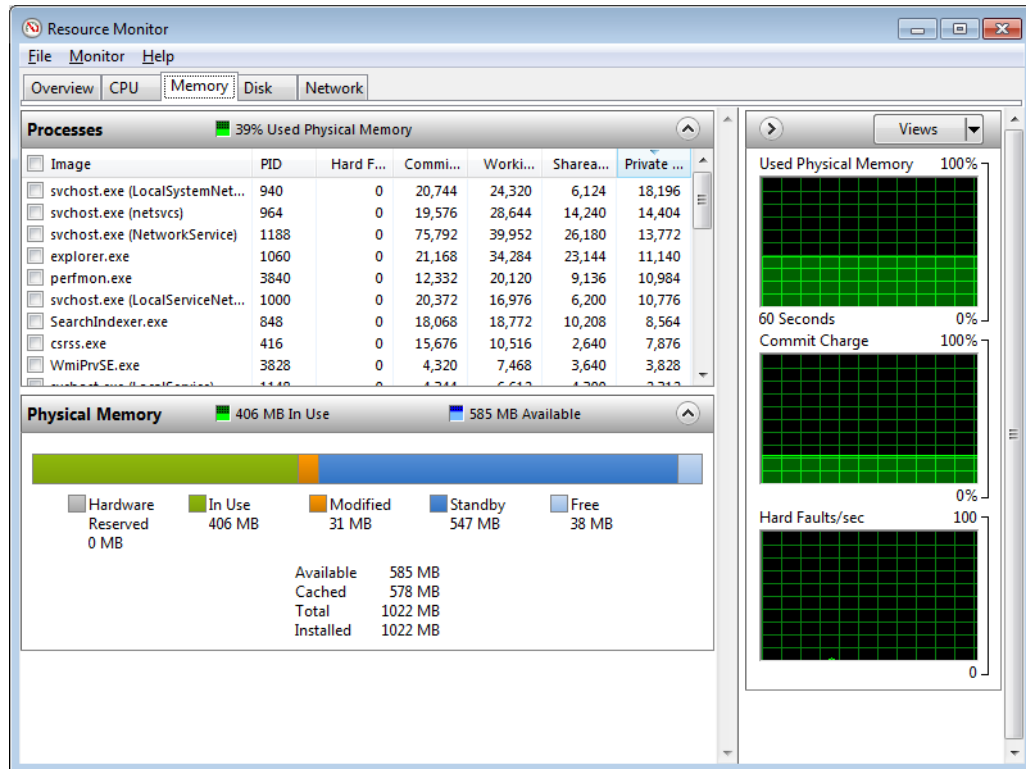
---

Windows implements a virtual memory system to allocate memory to components and processes. The virtual memory system uses the physical memory that is installed on the system together with a page file, which consists of one or more disk files. This allows Windows to give each process its own large memory address space.

The details of memory manager operations are beyond the scope of this paper. "Resources" at the end of this paper provides a number of references to help you develop a better understanding of the inner workings of the memory manager.

## Displaying Memory Usage by Using Resource Monitor

Resource Monitor is a built-in Windows system tool that displays information about the use of hardware resources, such as CPU, memory, disk, and network, and software resources such as file handles and modules in real time.



**Figure 1. Resource Monitor > Memory Tab**

Figure 1 shows the **Memory** tab of Resource Monitor. This tab shows three panes that have the following information:

- The upper-left pane shows a per-process breakdown of memory usage, including the following:
  - The average number of hard-faults per second that occurred in the last minute.
  - The current commit size.
  - The working set size of the process.
  - The size of shareable pages in the process working set.
  - The size of private and non-shareable pages in the process working set.
- The lower-left pane displays system-wide use of physical memory.
- The right pane shows memory use on the system over time.

The following sections describe the important categories of memory that Resource Monitor displays. For detailed information about Resource Monitor, see “Performance and Reliability” on TechNet.

## Commit Charge

Commit Charge is the total amount of memory that the memory manager has committed for a running system component or process. This number includes the amount of RAM that is in use and any modified pages that have been saved in the page file. The upper-left pane of the **Memory** tab shows the Commit Charge for each process as “Commit (KB)”, and the right pane shows the Commit Charge for the system over time.

## Process Working Set

The working set of a process is the subset of its virtual address space that is in use at a given time. A process’s working set can be divided into two main categories: shareable pages and private pages. The upper-left pane in Resource Monitor displays the size of the working set for each process as “Working Set (KB)”.

### Shareable Pages

Shareable pages are memory-resident pages that can be shared with other processes. One copy of the page is physically in memory and is mapped into the virtual address space of one or more processes, as required. Examples of shareable pages include executable image files and memory-mapped files.

System DLLs such as Ntdll, Kernel32, Gdi32, User32, Advapi, Commctl32, and Msvcrt are shareable. This optimization saves memory space and benefits most processes because only one copy of the page is required.

### Private Pages

Process private pages—also called the private working set—are the resident, non-shareable memory pages that are currently mapped into a process’s virtual address space. Resource Monitor shows the size of the private working set for each process. This metric is typically used to measure the memory impact of an application. However, this number does not show peak memory use or memory that is used for the application but is not reflected in the memory manager working set data structures. For example, the private working set does not include cached file access, memory in services that the application calls, shared pages, or shareable pages.

Examples of process private pages are:

- Pages on the process heap, common language runtime (CLR) heap, C runtime heap, and so on.
- Private memory allocations that the process requests by calling **VirtualAlloc**.
- Pages that are part of the user stack that the system maintains for every thread.
- Pages that were marked as copy-on-write and have since been copied.

## Pages Not Included in a Process Working Set

Many pages are indirectly accessed for a process and do not appear in its working set. The following are the most important pages:

- Pages that are maintained in kernel memory, such as data in paged and non-paged pools, kernel stacks, pages that are part of the hardware application layer (HAL), and pages that are part of Win32k.sys.

- Pages that other processes, such as a spooler, access for this process.
- Files that the process reads by using buffered read APIs such as **ReadFile**.

For more information, see “Working Set” on MSDN®.

## Trimming a Process’s Working Set

The memory manager monitors the working set of each process and the requests for additional memory resources. As the working set grows, the memory manager balances the process’s demand for memory against requests from the kernel and other processes. In some situations, the memory manager must trim the working set—that is, it must remove one or more pages. The memory manager trims a process’s working set if any of the following occurs:

- The system does not have enough memory available to satisfy current requests.
- The memory manager cannot expand the working set, but must add a new page to it. This typically occurs when the working set has reached its hard limit, if it has such a limit.
- The process explicitly reduces or empties its working set by calling one of the **SetProcessWorkingSet** functions.

If a system has a substantial amount of available memory—typically at least several hundred megabytes (MB)—the memory manager stops trimming process working sets. On systems that have more available memory, the memory manager trims the process working sets less aggressively. Therefore, if installed memory can accommodate much of the working set, Resource Monitor reports greater values for the In Use metric. Because more of the less frequently accessed pages of the working set can remain in memory, less frequent paging is required.

Trimming the working set does not remove the pages from physical memory. Instead, the memory manager moves the page to the Standby page list or the Modified page list. These lists are described in “Global Memory Usage” later in this paper.

## Page Faults

A page fault occurs when a process accesses a page of memory that is not currently in its working set. A hard page fault requires page contents to be retrieved from the disk, and a soft page fault can be resolved without accessing the disk.

A hard page fault occurs when a process accesses a page that is not resident in memory. This can happen if:

- An accessed page contains process private data that must first be read from the page file. Multiple hard faults from the page file typically require additional time so that the system can read pages from the disk. Frequent disk reads can decrease system responsiveness.
- Pages that contain sharable pages such as code or file data must be read from a file.

Soft page faults typically occur if:

- The page is in the working set of some other process, so it is already resident in memory.

- The page has either been removed from the working sets of all processes that are using the page and has not been given to another process or it is already resident because of a memory manager prefetching operation. The page is moved from the Standby or Modified list to the process's working set.
- A process references an allocated virtual page for the first time.

## Global Memory Usage

The memory manager tracks pages that are not part of the working set in one of four lists:

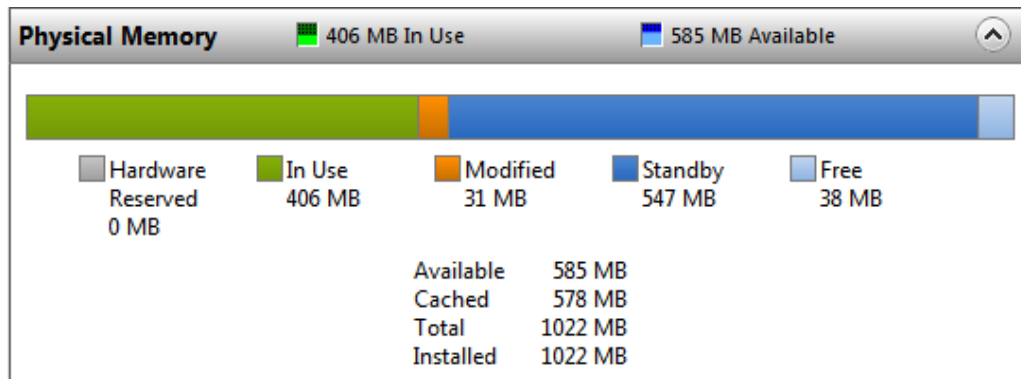
- Zero page list
- Modified page list
- Standby page list
- Free page list

These lists are system wide, so that the memory manager can balance the memory needs of the processes and components that are running on the machine.

Resource Manager shows the size of each list as follows:

- Modified Memory is the total amount of memory on the Modified page list.
- Standby Memory is the total amount of memory on the Standby page list.
- Free Memory is the total amount of memory on both the Zero page and Free page lists.

Figure 2 shows the Resource Manager display of the Modified, Standby, and Free page list sizes.



**Figure 2. Resource Monitor > Memory Tab > Physical Memory Display**

### In Use Memory

In Use memory is calculated by adding the sizes of the Modified, Standby, and Free list and subtracting this from the amount of recognized memory. Recognized memory is calculated by subtracting any Hardware Reserved memory from the installed RAM. For more information see "RAM: Installed versus Recognized" later in this paper.

### Zero Page List

The Zero page list contains pages that have been initialized to zero, ready for use when the memory manager needs a new page.

**Modified List**

The Modified page list contains pages that have been modified but have not been accessed for some time and have been removed from a process working set. After a modified page has been written to the page file, the memory manager moves the page to the Standby page list.

If the memory manager detects an increase in memory pressure, it arranges for modified pages to be written to disk, which makes previously modified pages available for repurposing.

After the memory manager initially writes out several pages to the page file, it checks whether the system is still under memory pressure. If so, the memory manager quickly writes out more pages. If the system is no longer under memory pressure, the memory manager writes the modified pages to disk slowly, in the background. This minimizes risk from data integrity issues that might occur if modified pages remain in memory.

**Standby List**

The Standby list contains unmodified pages that have been removed from process working sets, which effectively makes the Standby list a cache. If a process needs a page that is on the Standby list, the memory manager immediately returns the page to its working set.

All pages on the Standby list are available for memory allocation requests. If a process requests memory, the memory manager can take a page from the Standby list, initialize it, and allocate it to the calling process. This is called repurposing a page.

Pages on the Standby list are often from recently used files. By keeping these pages on the Standby list, the memory manager reduces the need to read information from the disk. Disk reads can decrease system responsiveness.

**Free Page List**

The Free page list tracks memory pages that have not been allocated for a purpose or were previously allocated and returned to the memory manager for reuse.

The memory manager maintains a thread that wakes up periodically to initialize pages on the Free page list so that it can move them to the Zero page list.

## System Reference Set

The size of the working set is the traditional way to measure the memory impact of a process. However, the working set size is a flawed metric because it does not show complete memory usage. It does not show all the memory that is used to support the process, such as cached files, or memory in services that the process calls.

The reference set is the system-wide set of pages that any process or activity on the system has accessed. Reference set is an extremely accurate way to assess the memory usage across the system for any workload because pages that are currently in the working set of any process, or that are accessed by system processes or drivers, count toward the reference set.

The Windows 7 development team used an instrumented version of the memory manager to track which memory pages were accessed and by which process, as well as details about the type of information that is stored in the page (code, data, or file contents). The instrumentation enabled the development team to quantify the required number of physical memory pages to support a workload that is independent of installed RAM, memory manager, or system service policies.

## Results of Memory Use Analysis on Sample Systems

---

The amount of memory that an idle system uses depends on the system's hardware configuration, drivers, and the services and processes that are running. Therefore, two systems that have different hardware, drivers, or amounts of RAM can use dramatically different amounts of memory.

For example, Resource Monitor reports more memory as In Use on a system that has 3 gigabytes (GB) of RAM than on an identically configured system that has 1 GB of RAM. On a system that has more RAM, Windows makes several adjustments to use the additional memory, including the following:

- Using larger control structures to manage the extra memory.
- Making additional non-paged pool reservations at system boot time.
- Allowing system services to use more memory.

On systems that have more installed memory, the memory manager is less aggressive in trimming process working sets, which causes Resource Monitor to report higher values for the In Use metric. Specifically, tools such as Resource Monitor and Task Manager report higher amounts of RAM in use and greater values for both process working sets and private working sets when the system is not under memory pressure.

As described in "Reference Set" earlier in this paper, we measured the size of the reference set during Windows 7 development.

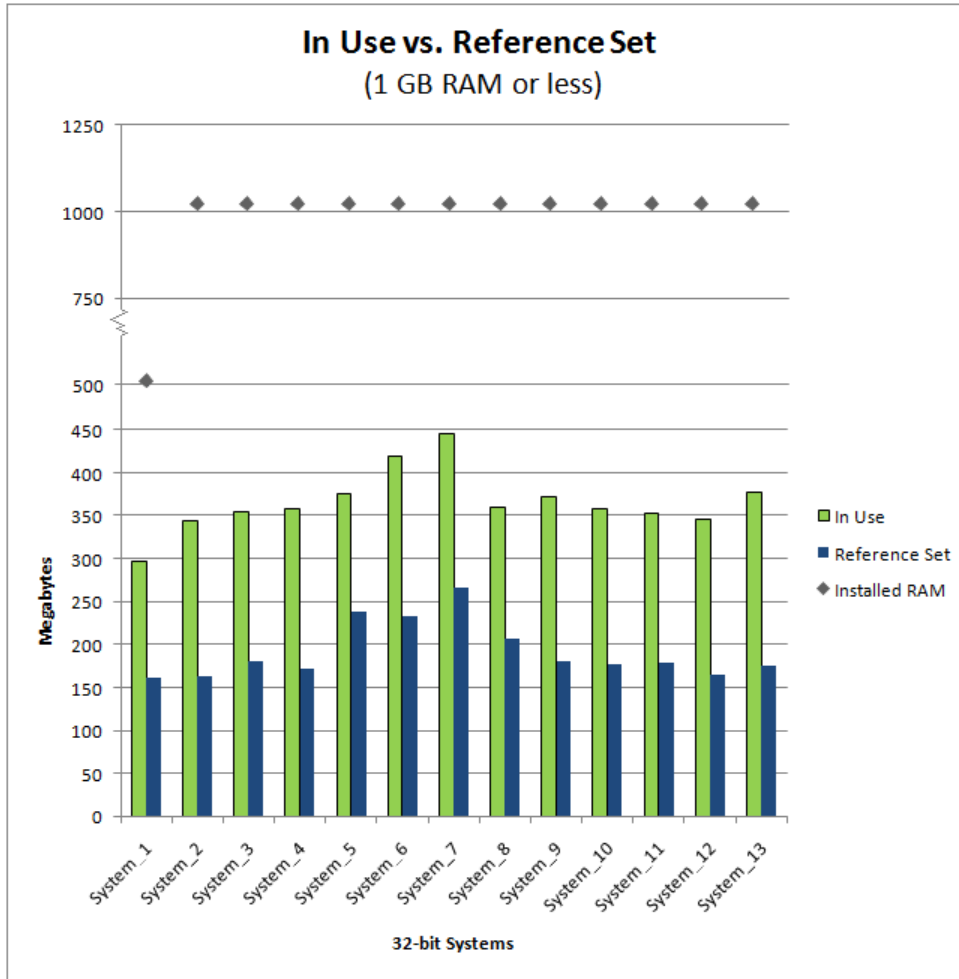
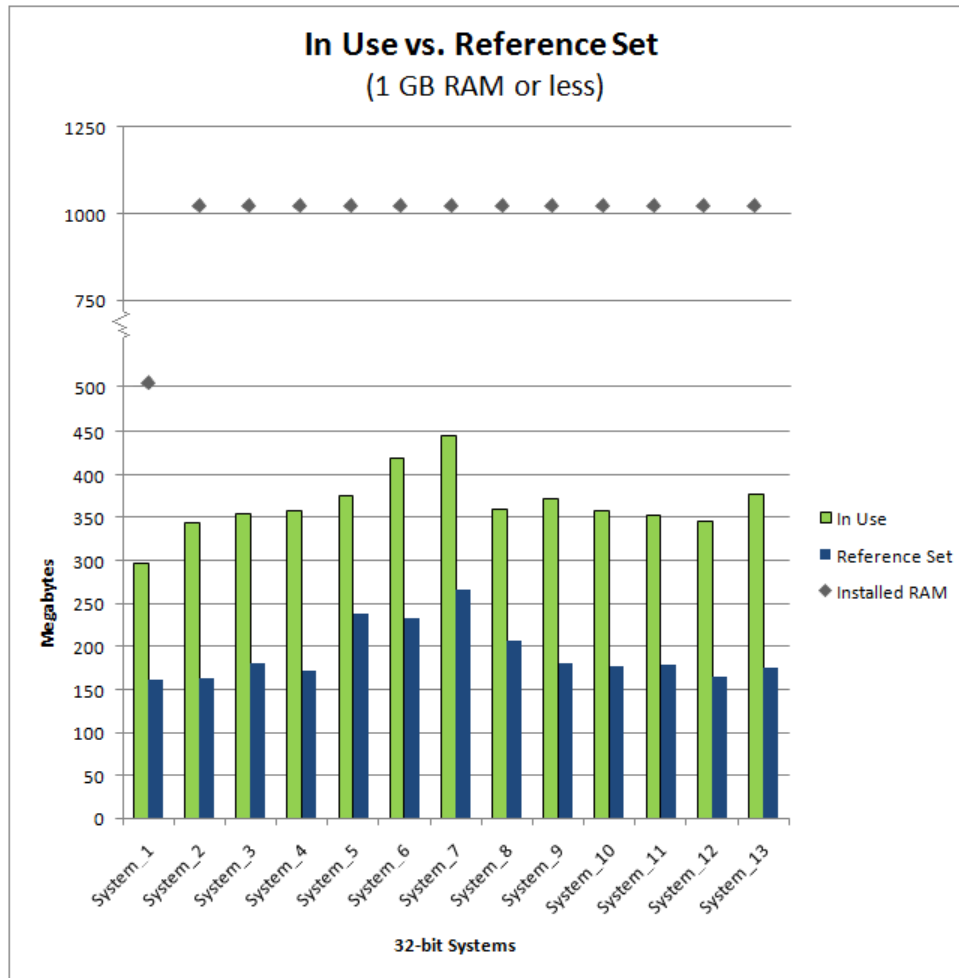


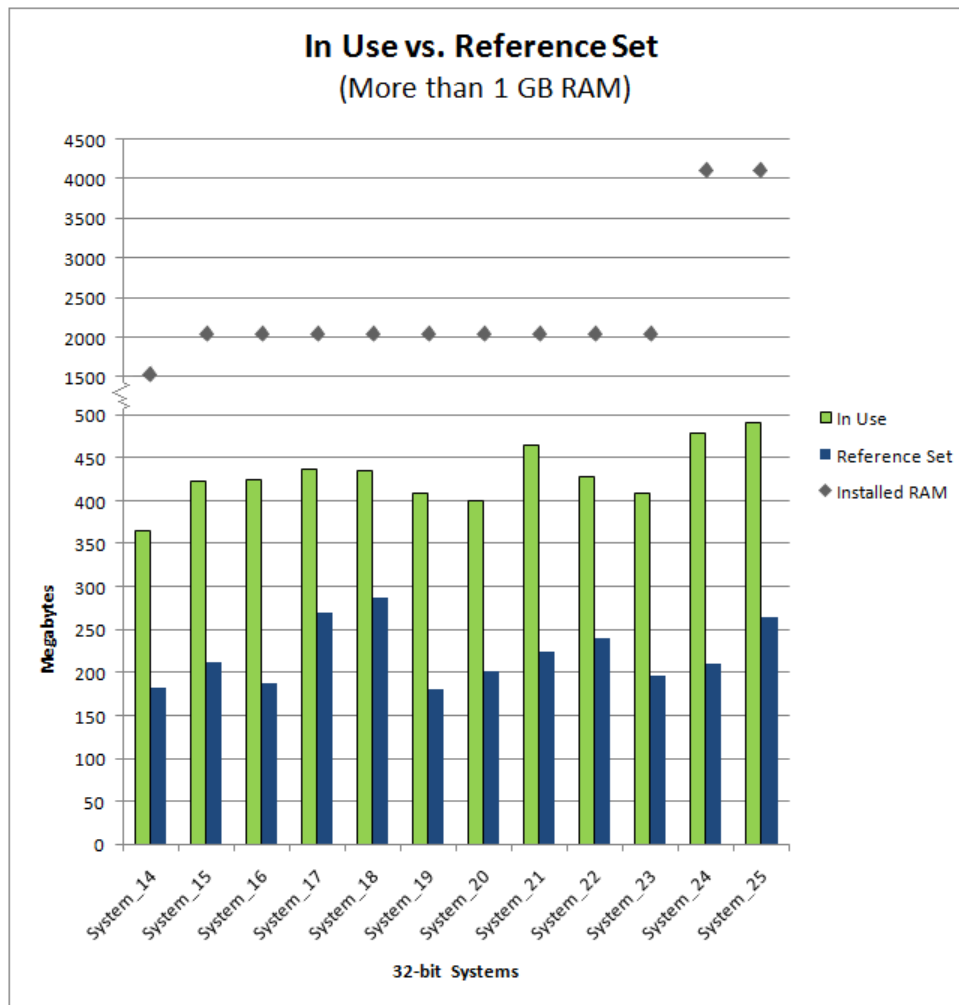
Figure 3 shows the amount of In Use memory that Resource Monitor reports, compared with the size of the reference set over a 10-minute period, measured 20 minutes after boot for 32-bit Windows 7.



**Figure 3. In Use memory and reference for 32-bit systems (1-GB RAM or less)**

System\_6 and System\_7 report In Use figures that are higher than other similar systems. This is because both systems allocate RAM for the Unified Memory Architecture (UMA) graphics subsystem because they have no dedicated video memory. For more information on tested configurations, see “Appendix A. Tested Configurations” later in this paper.

Figure 4 shows the same data for 32-bit systems with more than 1-GB RAM.

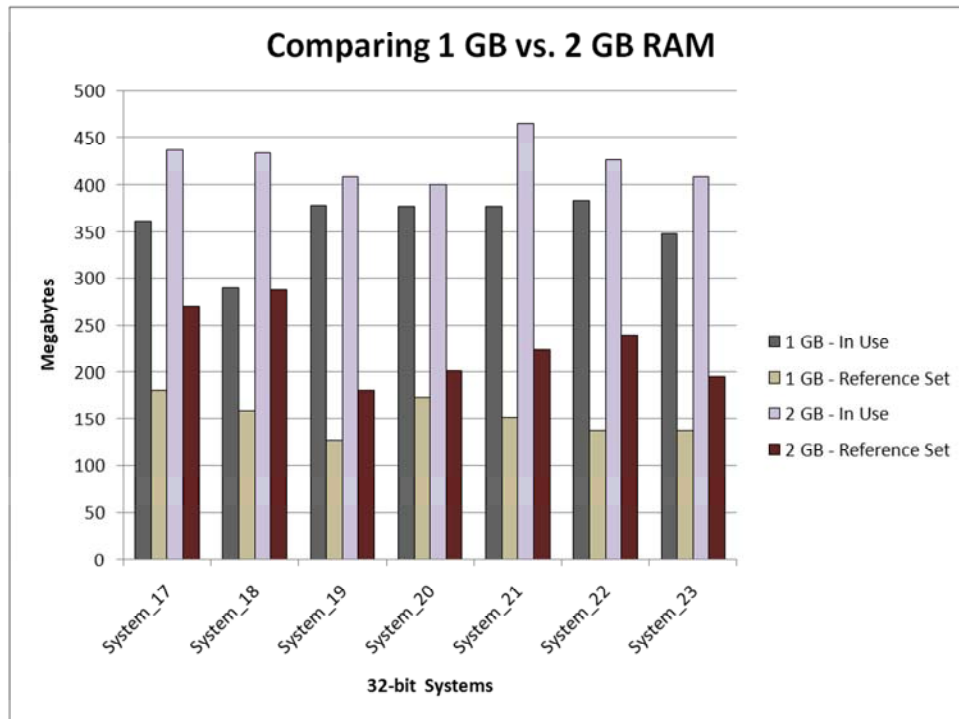


**Figure 4. In Use memory and reference set for 32-bit systems (more than 1-GB RAM)**

Figure 3 showed that 1-GB systems report 300 to 450 MB of memory in use, which indicates that several hundred MB of memory is available on the Free and Standby lists.

Figure 4 shows that the In Use and Reference Set values are slightly greater for systems with more than 1 GB of RAM. Because so much memory is still available, the memory manager is conservative about removing pages from process working sets.

Figure 5 shows the effect of different memory configurations on both In Use and Reference Set values.

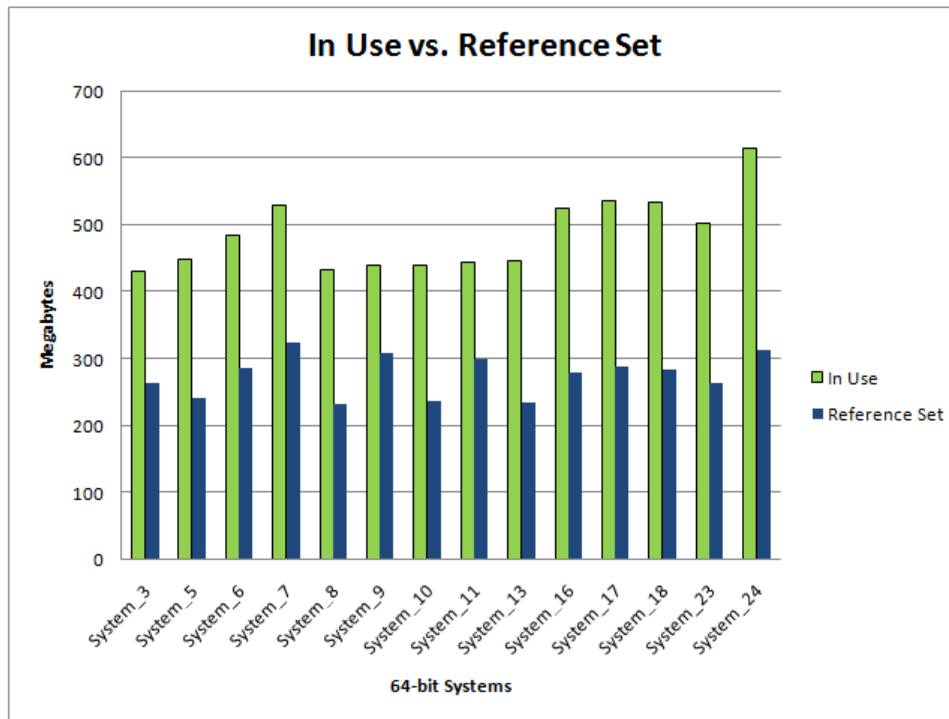


**Figure 5. Reference set for the same system with 1-GB and 2-GB RAM**

The In Use and Reference Set values differ on the same system, when memory increases from 1 GB to 2 GB, for the following reasons:

- Windows adjusts some system policies when more RAM is available.
- The internal structures that Windows uses to manage the additional memory require additional memory themselves.
- The memory manager is conservative about removing pages from process working sets when more memory is available.

Figure 6 shows the amount of memory In Use and Reference Set for Windows 7 on 64-bit Windows, when running the same memory tests on the same systems with x64 CPUs.



**Figure 6. In Use memory and reference set for 64-bit systems**

Figures 7 and 8 categorize the Reference Set data from Figures 2 and 3, respectively, to show how the system uses memory. These figures show the following categories:

- File  
File contents.
- Code  
Code or import tables and modified static data.
- Stack  
User-mode or kernel-mode stacks.
- Dynamic  
Dynamic allocations, such as heap memory that the processes allocate.
- Paged Pool  
System allocations as described earlier.
- Non-Paged Pool  
System allocations as described earlier.
- System  
Driver and other memory pages that Windows uses to track system state.

The memory manager reference set instrumentation provides additional details that you can use to determine the source of all memory usage on a system at a particular time. By using this data, the development team determined that on a clean

installation of Windows 7, pool allocations and driver code pages are the largest consumers of memory when the system is idle.

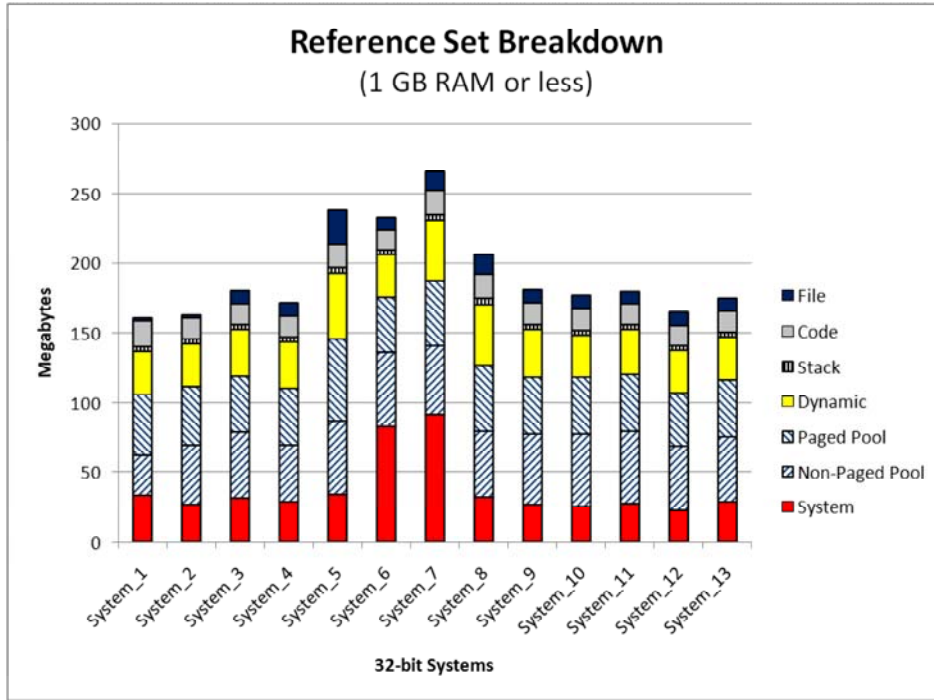


Figure 7. Reference set by category for systems with 1 GB or less of RAM

System\_6 and System\_7 have larger System memory sizes because they do not have dedicated video memory. On these systems, Windows allocates 57.5 MB of RAM for the UMA graphics subsystem.

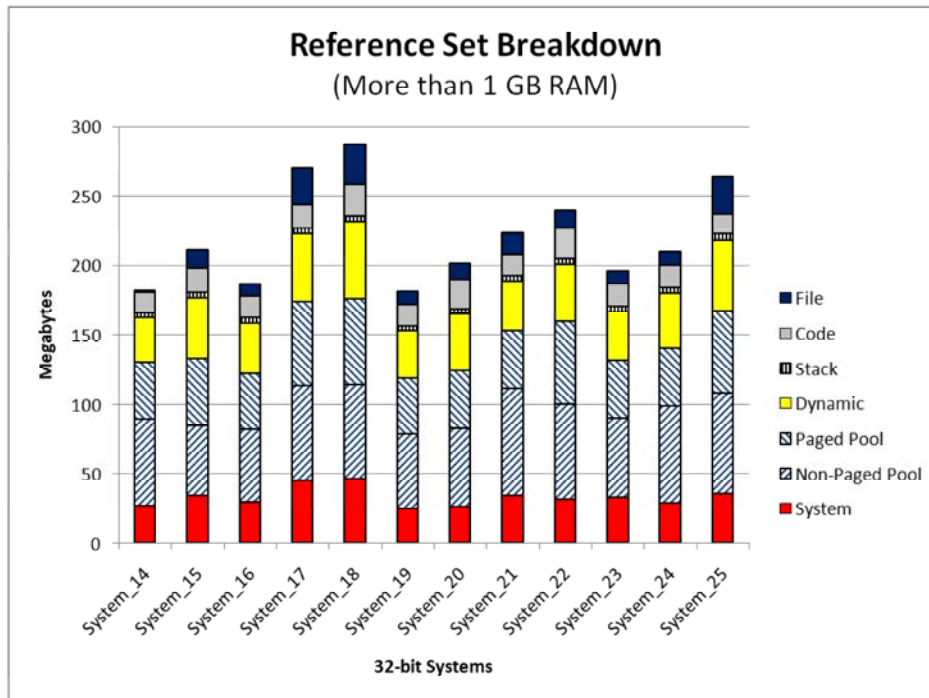


Figure 8. Reference set by category for systems with more than 1 GB of RAM

As users install additional software and hardware on a system, the amount of memory in use increases even when the installed software is not active. The reasons for the increase include the following:

- Additional driver code pages and driver pool allocations.
- Additional processes and threads.
- Additional code pages for additional features, such as automatic file association.
- More cached pages on the Standby list to reduce application startup latency.
- Additional memory for system user interface components such as the Start menu.

Figure 9 compares the reference set of an idle, clean Windows 7 installation to the same idle system with additional software and hardware installed. The installed software includes Microsoft® Office, Adobe Flash, Zune®, and a digital camera support package. The additional hardware includes a multifunction printer and an external hard drive for archived files. These additions are responsible for a 14-percent or 23-MB increase in idle system reference set, with the largest increases occurring in pool and dynamic allocations.

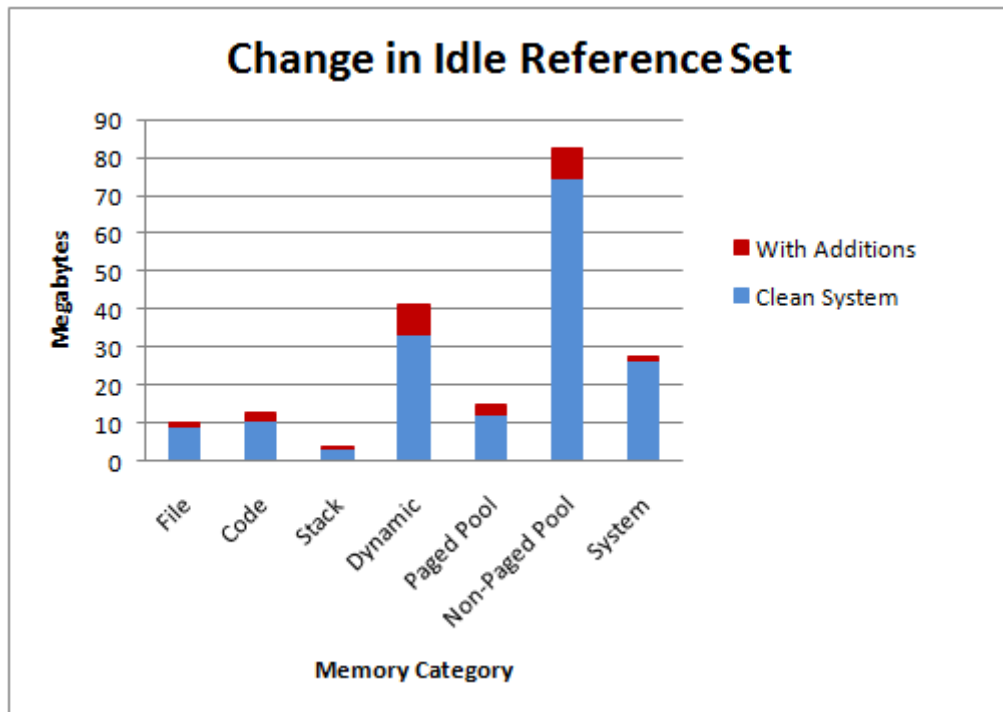


Figure 9. Change in idle reference set after additional installations

---

## Determining the Required Memory for a Configuration

---

Performance testing requires many resources and a great deal of time, but it is a critical step to ensure that the system configuration has adequate memory to perform as expected for specific usage scenarios.

Testing Windows 7 can be difficult because the operating system tries to improve its performance over time based on observed usage patterns. The best way to ensure good results is to perform each test multiple times by following the same steps and methodology. In addition, by testing for longer periods of time, you can capture activities that increase memory use but occur infrequently or for a short duration.

### Methodology Background

To determine how much memory a particular user scenario requires, we recommend that you monitor the number and priority of the pages on the Standby list.

Windows 7 SuperFetch™ works with the memory manager to set priorities for pages on the Standby list. Pages that have recently been placed on the Standby list start with a high priority, which slowly decreases over time. If the system has too little memory to handle the workload, only a small percentage of the pages on the Standby list have a low priority. By monitoring the number of pages at different priorities, you can determine whether the system would benefit from increasing the amount of installed RAM.

Overall, the goal is to ensure that the system can maintain recently referenced pages in memory and still have enough available memory to satisfy any immediate memory needs.

The following sections provide some guidance on how to determine the appropriate amount of memory to install on a system.

### Testing the Configuration

To test the memory performance of a configuration, follow these steps:

1. Configure the hardware on the test system.
2. Configure the software on the test system.
3. Create a test script.
4. Configure data capture tools.
5. Run the test.

#### Step 1: Configure the Hardware

##### **To configure the hardware on the test system**

---

Set up the test system with the same hardware configuration that you plan to include in a shipping system. Use the same amount of RAM, video adapter, disks, network cards, and so on when doing performing testing.

## Step 2: Configure the Software

### To configure the software on the test system

---

1. Install and activate the installation of Windows 7.
2. Install the latest drivers.
3. Install all required patches and fixes from Windows Update and turn Windows Update off.
4. Install any applications that you plan to provide as part of the standard image, such as anti-virus programs, firewalls, value-added applications, search tools, Internet browsers, browser plug-ins (such as Flash and Search Plug-Ins), and so on.
5. Install the applications that constitute the workload for your testing scenario. For example, to test a system that is intended for a graphics-heavy workload, install the graphics applications that your customers are most likely to use.
6. If appropriate, connect the system to the appropriate network, ensuring that you select the most representative network type (Home, Work or Public). If you expect the end-user's machine to be part of a domain, join your test machine to a domain of similar size and network traffic.

## Step 3: Create a Test Script

Create a test script that:

- Allows the system to be idle for 20 minutes after boot. This reduces variance in the test results that SuperFetch can cause.
- Uses the applications in the same order that users would.
- Performs the testing tasks for a specified frequency and duration.

To produce a better picture of the end-user experience, repeat the same test at least five times.

## Step 4: Configure Data Capture Tools

### To configure data capture tools

---

Use Performance Monitor (PerfMon) to capture information about memory usage. Manually measure the response times for the various workload actions that occur during testing to compare them against additional tests that you run on the same and different memory configurations.

Record the following performance counters:

- Memory – Standby Cache Reserve Bytes  
These pages have not been in the working set of a process for a long time and therefore have a low page priority. The memory manager leaves pages at this priority level for a long time unless it requires additional pages to satisfy a demand for memory. The number of pages at this priority level indicates the long-term memory demands of the system workload.

- **Memory – Free & Zero Page List Bytes**  
This value is the total number of types of memory on the Free and Zero page lists. Systems that have large amounts of memory in this category have more memory than is required for the current workload.
- **Memory – Pages Input/Sec**  
This value is the number of memory pages that the system reads from disk each second because of hard page faults. These pages can come from the page file, executable images, or other files.

### Step 5: Run the Test

Run the test scenario several times on each hardware configuration. Record test results and PerfMon logs for later analysis.

## Analyzing the Results

After testing is complete, analyze the data from each memory configuration and scenario. Consider both quantitative and qualitative criteria. Quantitative criteria include the values of performance counters that you recorded with PerfMon. Qualitative criteria are related to end-user expectations and typically include:

- System responsiveness.
- Time to complete a scenario.
- Time to move between application screens.

The optimal memory size for a system depends on the hardware and software that are installed and the workload that the system typically runs. Often, the end-user's perception of the system responsiveness under load determines the minimal memory requirements, but several performance counters can also indicate that a system is under memory stress.

Table 1 lists the performance counters to evaluate during testing. The two most important are Standby Cache Reserve Bytes and Free & Zero Page List Bytes. The sum of these two counters should remain above the suggested thresholds shown in the table during the test. The Pages Input/Sec is critical if the sum of these counters is below the suggested threshold. In these cases, little or no paging indicates that the system has enough memory to contain the workload, but the workload frequently accesses a large amount of memory. Frequent accesses prevent the memory manager from moving pages to the Standby list or reducing their priority, but this also means that the workload fits entirely in memory. In this situation, additional RAM would be of marginal benefit.

**Table 1. Memory Performance Counters**

Performance counter	Suggested threshold value
Memory – Standby Cache Reserve Bytes	<ul style="list-style-type: none"> <li>▪ Systems with 1-GB RAM: Sum of Standby Cache Reserve bytes and Free and Zero page list bytes is at least 200 MB.</li> <li>▪ Systems with 2-GB or more RAM: Sum of Standby Cache Reserve bytes and Free and Zero page list bytes is at least 300 MB.</li> </ul>

Performance counter	Suggested threshold value
Memory – Free & Zero Page List Bytes	<ul style="list-style-type: none"> <li>▪ Systems with 1-GB RAM: Sum of Standby Cache Reserve bytes and Free and Zero page list bytes is at least 200 MB.</li> <li>▪ Systems with 2-GB or more RAM: Sum of Standby Cache Reserve bytes and Free and Zero page list bytes is at least 300 MB.</li> </ul>
Memory – Pages Input/Sec	Average over a 1-hour period is less than 10

Occasional short-term transitions above or below the thresholds are acceptable because they indicate the system's peak demand. A peak demand that crosses the threshold can diminish performance for several minutes after the event.

If the sum of Standby Cache Reserve Bytes and Free & Zero Page List Bytes is below the suggested threshold and the Pages Input/Sec is above the threshold, the workload is likely to perform better if the system has more RAM. To confirm the level of benefit, repeat the test with additional memory.

## Best Practices for Memory Performance Testing

The following are best practices for memory performance testing:

- Test a representative configuration.
  - To obtain useful test results, the tested system must accurately represent the end-user configuration. The installed hardware and software should be as close as possible to the system that you plan to ship.
- Automate the testing.
  - To test the impact of different memory configurations, use a script or other automated process to ensure consistency. Automated testing enables you to run the same scripted scenario under several different memory configurations.
- Minimize the effect of monitoring.
  - Performance testing itself changes the behavior of the system that is being tested. Tools that capture and record performance information change the amount of memory that is available to run the test scenario.

## Resources

### Books

***Microsoft Windows Internals*, Mark E. Russinovich, David A. Solomon**  
 Microsoft Press  
 ISBN 978-0-7356-2530-3

### WHDC Web site

#### Advances in Memory Management for Windows

<http://www.microsoft.com/whdc/system/sysinternals/MemMgt.aspx>

#### Graphics Guide for Windows 7

<http://www.microsoft.com/whdc/device/display/GraphicsGuideWin7.aspx>

#### Window Performance

<http://www.microsoft.com/whdc/system/sysperf/default.aspx>

**Windows Driver Kit**

<http://www.microsoft.com/whdc/ddk/winddk.aspx>

**Microsoft Assessment and Planning Toolkit**

<http://www.microsoft.com/downloads/details.aspx?FamilyID=67240b76-3148-4e49-943d-4d9ea7f77730&DisplayLang=en>

**MSDN**

**Memory Limits for Windows Releases**

<http://msdn.microsoft.com/en-us/library/aa366778.aspx>

**Memory Management**

[http://msdn.microsoft.com/en-us/library/aa366779\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/aa366779(VS.85).aspx)

**PoolMon**

<http://msdn.microsoft.com/en-us/library/ms792884.aspx>

**Working Set**

[http://msdn.microsoft.com/en-us/library/cc441804\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/cc441804(VS.85).aspx)

**Knowledge Base Articles**

**The system memory that is reported is less than you expect if 4 GB of RAM is installed**

<http://support.microsoft.com/?scid=kb%3Ben-us%3B929605&x=13&y=19>

**Microsoft TechNet**

**Monitoring System Activity with Resource View**

[http://technet.microsoft.com/en-us/library/cc766086\(WS.10\).aspx](http://technet.microsoft.com/en-us/library/cc766086(WS.10).aspx)

**Performance and Reliability Monitoring Step-by-Step Guide for Windows Server 2008**

[http://technet.microsoft.com/en-us/library/cc771692\(WS.10\).aspx](http://technet.microsoft.com/en-us/library/cc771692(WS.10).aspx)

**Performance Monitoring Getting Started Guide**

[http://technet.microsoft.com/en-us/library/dd744567\(WS.10\).aspx](http://technet.microsoft.com/en-us/library/dd744567(WS.10).aspx)

**Windows Sysinternals**

<http://technet.microsoft.com/en-us/sysinternals/default.aspx>

**VMMMap v2.1**

<http://technet.microsoft.com/en-us/sysinternals/dd535533.aspx>

## Appendix A. Tested Configurations

Table A-1 summarizes the tested hardware configurations.

**Table A-1. Tested Systems Hardware Summary**

Name	Form factor	RAM		Graphics		Network cards	
		Installed	Recognized	UMA	Discrete	LAN	Wi-Fi
System_1	Desktop	512	446	X		2	
System_2	Notebook	1024	895	X		1	1
System_3	Desktop	1024	958	X		1	
System_4	Desktop	1024	959	X		1	
System_5	Desktop	1024	1005		X	2	
System_6	Desktop	1024	1014	X		1	
System_7	Desktop	1024	1014	X		1	
System_8	Desktop	1024	1022		X	1	
System_9	Desktop	1024	1022		X	1	
System_10	Desktop	1024	1022		X	1	
System_11	Desktop	1024	1023		X	1	
System_12	Desktop	1024	1023		X	1	
System_13	Desktop	1024	1023		X	1	
System_14	Notebook	1536	1535		X	1	1
System_15	Desktop	2048	1983	X		1	
System_16	Desktop	2048	1983	X		1	
System_17	Desktop	2048	2046		X	1	
System_18	Desktop	2048	2046		X	1	
System_19	Desktop	2048	2046		X	1	
System_20	Desktop	2048	2046		X	1	
System_21	Notebook	2048	2046		X	1	1
System_22	Desktop	2048	2047		X	3	
System_23	Desktop	2048	2047		X	2	
System_24	Desktop	4096	3326		X	1	
System_25	Desktop	4096	3327		X	1	

### RAM: Installed versus Recognized

The difference between the installed and recognized amounts of RAM is usually because of hardware reservations. When hardware components reserve memory, the memory manager cannot use certain memory address ranges.

Reservations are typically 10 to 70 MB of memory, but they vary depending on the machine's hardware configuration and might be several hundred MB. Examples of components that can affect the amount of memory reserved include:

- System BIOS.
- Motherboard resources, such as I/O advanced programmable interrupt controller (APIC).
- Any device that requires memory-mapped I/O, such as a sound card.
- PCI Express (PCIe) configuration space.
- Any additional PCI device memory.

- Video graphics array (VGA) memory for video aperture.
- Other chipset and system devices, such as flash devices.

## UMA Graphics Hardware

Some graphics hardware uses a Unified Memory Architecture (UMA), which is sometimes called Shared Memory Architecture (SMA). In UMA systems, the graphics subsystem does not have its own dedicated memory. Instead it uses some of the installed RAM, sharing this memory with the operating system. In contrast, discrete graphics hardware includes dedicated graphics memory.

Other consequences of UMA graphics hardware, compared to discrete graphics hardware, can include the following:

- Slower graphics performance because system memory is generally slower than dedicated graphics memory.
- More contention on the memory bus because the graphics hardware must coordinate access to the installed memory with the remainder of the operating system.

## Appendix B. Tools for Measuring Memory Usage

---

Although no tool in Windows 7 can measure reference set, several tools provide information about working sets of processes and system-wide memory usage:

- Windows Task Manager
- Windows Resource Monitor
- Windows Performance Monitor

In addition, the Windows Driver Kit (WDK) and third parties provide tools that can help you assess memory usage:

- PoolMon, from the WDK. For more information, see “PoolMon” on MSDN, which is listed in “Resources.”
- VMMap, from SysInternals. For more information, see “VMMap v 1.1” on Microsoft TechNet, which is listed in “Resources.”