|  |
| --- |
| Microsoft |
| Remote Desktop Protocol Performance |
|  |
| **Presentation and Hosted Desktop Virtualization Team** |
| **10/13/2008** |

Contents

[Overview 3](#_Toc211656446)

[User Scenarios 3](#_Toc211656447)

[Test Setup 4](#_Toc211656448)

[Remote Desktop Connection Settings 4](#_Toc211656449)

[Presentation Virtualization 4](#_Toc211656450)

[Color Depth 4](#_Toc211656451)

[ClearType Virtualization (Font Smoothing) 5](#_Toc211656452)

[Bitmap Caching 7](#_Toc211656453)

[Desktop Themes 8](#_Toc211656454)

[Desktop Composition 9](#_Toc211656455)

[RemoteApp Programs 10](#_Toc211656456)

[Bulk Compression 11](#_Toc211656457)

[Performance Comparison with Earlier RDP Versions 13](#_Toc211656458)

[Bandwidth Allocation for Terminal Server Connections over RDP 15](#_Toc211656459)

[Device Redirection 16](#_Toc211656460)

[Audio 16](#_Toc211656461)

[Printer 16](#_Toc211656462)

[Conclusion 17](#_Toc211656463)

# Overview

With the release of the Windows Server® 2008 and Windows Vista® operating systems, Remote Desktop Protocol (RDP) is more feature rich, enabling new presentation and remote-oriented features such as Desktop Window Manager (DWM) presentation virtualization, 32-bit support, ClearType® display technology, and device redirection.

As these features become integrated in the enterprise environment, it is important to analyze and understand the impact on your current network infrastructure and the end-user experience. This paper details different RDP features and the potential impact to usability and bandwidth.

To test the impact of different features and compare RDP 6.1 to previous versions, we performed a variety of tests by using automated and simulation tools to demonstrate the user scenarios outlined in this white paper. These tests simulate a user working with actual applications at realistic speeds. This simulation provides a more reliable benchmark than tools that replay pre-recorded graphics data at high speeds (which creates a less realistic testing environment).

# User Scenarios

Five different user scenarios were used to measure the performance of the Remote Desktop Connection (RDC) client.

* **Executive PPT Scenario.** This scenario emulates a user presenting 28 high-fidelity slides by using Microsoft® Office PowerPoint® 2007. The slides contain images, transitions, and backgrounds with color gradient. The user spends about 20 seconds on each slide.
* **Simple PPT Scenario.** This scenario emulates a user creating and presenting content by using PowerPoint 2007. The slides in this scenario are more text-intensive than those in the executive PowerPoint scenario and have a plain background. Some of the slides contain digital photo images.
* **Typing and Scrolling Scenario.** This scenario emulates a user scrolling through a 10-page document and typing 8 pages in another document by using Microsoft Office Word 2007. The user types at 35 words per minute and scrolls at about 2 pages per minute. The user also moves and resizes the windows as he moves between the two documents.
* **Scrolling Scenario.** This scenario emulates a user scrolling through a 275-page Word 2007 document that contains several Microsoft Office Visio® 2007 (data-flow) diagrams and embedded tables. The user scrolls at about 2 pages per minute.
* **Internet Explorer Scenario.** This scenario emulates a user browsing the Web by using Internet Explorer 7. The user browses and scrolls through multiple Web pages that contain a mix of text, natural images, and some schematic diagrams. The Web pages are stored on the local disk drive of the terminal server to avoid errors due to varying load times.

Note: Data for each scenario is not available for each test case.

# Test Setup

The tests were conducted in a private lab to avoid external network interference. The results in this white paper are the average of two test runs.

## Remote Desktop Connection Settings

Tests were run by using the following Remote Desktop Connection client settings:

* Color depths: 8-Bit Color (8-bit) , High Color (15-bit), High Color (16-bit), and Highest Quality (32-bit)
* Connection speed: Modem (56 kilobits per second [Kbps])
* Default bulk compression settings, unless others are mentioned

# Presentation Virtualization

## Color Depth

With the release of Windows Vista, new user interface (UI) effects and practices that were previously uncommon scenarios for Terminal Services have been introduced. Applications that utilize the new Windows® Aero® desktop experience feature, including the desktop shell, Internet Explorer, and Microsoft Office 2007, use the alpha channel to render transparency/translucent effects. It is expected that many third-party applications will transition to using the same visual effects that are available in Windows Aero. To enable the full fidelity of the user interface of Windows Vista, RDP has added support for 32-bit color depth. To make 32-bit presentation virtualization more efficient, a new compressor for 32-bit bitmaps has also been added. Our internal test results below demonstrate that 32-bit presentation virtualization performs much better than 24-bit.

Note: The True Color (24-bit) option is no longer fully supported for Terminal Services in Windows Server 2008. The only valid options are High Color (16-bit) and Highest Quality (32-bit). If you select True Color (24-bit) in the Terminal Services Configuration snap-in or in the RDC client, the connection will default to High Color (16-bit).

Table 1 and Chart 1 compare the bandwidth (in kilobytes per second [KBps]) consumed by a user scenario that uses different color depth settings. The tests were conducted by using a 56-Kbps modem connection setting with the default bulk compression settings.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User scenario** | **32-bit** | **16-bit** | **15-bit** | **8-bit** |
| Executive PPT | 119.4 | 122.6 | 108.8 | 51.2 |
| Simple PPT | 56.8 | 50.5 | 45.3 | 20.2 |
| Typing and Scrolling | 20.2 | 2.8 | 2.8 | 1.7 |
| Scrolling | 2.2 | 0.9 | 0.9 | 0.7 |
| Internet Explorer | 90.47 | 19.71 | 20.84 | 4.79 |

**Table 1:** Color depth comparison at different bit rates

**Chart 1:** Color depth comparison for different scenarios

## ClearType Virtualization (Font Smoothing)

ClearType display technology is a Microsoft font smoothing technique that improves the readability of text on LCD screens. With the proliferation of LCD screens and the release of Windows Vista and Office 2007, ClearType has become very important. Most of the fonts available in Windows Vista and Office 2007 are tuned for ClearType and look unattractive when it is turned off. Due to these reasons, Terminal Services decided to give the user the option to turn on ClearType. You can get ClearType in the RDC 6.1 client by going to the **Experience** tab and selecting **Font smoothing**. However, the high fidelity of ClearType comes at a cost.

Normally (with font smoothing disabled) fonts are transmitted as glyphs. Remote Desktop Protocol transmits glyphs efficiently and caches them to reduce bandwidth consumption. With ClearType enabled, fonts are transmitted as bitmaps and not as glyphs. RDP does not transmit these bitmaps efficiently, resulting in increased bandwidth consumption. From our initial internal testing, we found that enabling ClearType for text editing or scrolling scenarios resulted in an increase in bandwidth consumption that was four to ten times greater than when the scenario was run with ClearType disabled.

Table 2 and Chart 2 show bandwidth consumption (in KBps) for user scenarios running at 16-bit color depth.

|  |  |  |  |
| --- | --- | --- | --- |
| **User scenario** | **ClearType text** | **Normal text** | **Bandwidth increase using ClearType** |
| Typing and Scrolling | 4.45 | 2.82 | 57.64 % |
| Scrolling | 3.18 | 0.88 | 260.92 % |
| Internet Explorer | 176.30 | 19.71 | 794.52 % |

**Table 2:** Bandwidth consumption for ClearType text versus Normal text at 16-bit color depth

**Chart 2:** Bandwidth consumption for ClearType text versus Normal text at 16-bit color depth

Table 3 and Chart 3 show bandwidth consumption (in KBps) for user scenarios running at 32-bit color depth.

|  |  |  |  |
| --- | --- | --- | --- |
| **User scenario** | **ClearType text** | **Normal text** | **Bandwidth increase using ClearType** |
| Typing and Scrolling | 21.89 | 20.20 | 8.377424382 |
| Scrolling | 9.90 | 2.24 | 341.0453135 |
| Internet Explorer | 463.794482 | 90.47 | 412.6289867 |

**Table 3:** Bandwidth consumption for ClearType text versus Normal text at 32-bit color depth

**Chart 3:** Bandwidth consumption for ClearType text versus Normal text at 32-bit color depth

Font smoothing is disabled by default on the RDC client, and can be enabled on the **Experience** tab. The RemoteApp Wizard enables font smoothing by default. For Windows Server 2008, there is no Group Policy setting to disable font smoothing.

## Bitmap Caching

The Remote Desktop Connection client supports both memory-based and persistent disk caches. Memory-based and persistent disk caches save the bitmaps from the server to the client computer in memory or on the disk; this allows cached bitmaps to be reused between client sessions and provides a much larger cache size. Caching saves about 25% bandwidth for most of our user scenarios. We have tuned cache settings to get the best performance out of the box. There may be cases where you need to modify the default cache sizes for your scenario. To modify the cache sizes, use either of the following:

* Registry key for persistent cache: **HKEY\_CURRENT\_USER\SOFTWARE\Microsoft\Terminal Server Client\BitmapPersistCacheSize**
* Memory cache size specified in the RDP file. For example, the following entry in the RDP file “bitmapcachesize:i:1500” will set the memory cache size to 1500 kilobytes.

Table 4 and Chart 4 show the bandwidth consumption (in KBps) for different user scenarios as we change the cache size. These tests were conducted at 16-bit color depth over a 56-Kbps modem.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Cache size** | **Executive PPT** | **Simple PPT** | **Typing and Scrolling** | **Scrolling** |
| No caching | 160.50 | 67.40 | 45.17 | 0.95 |
| 500-KB memory cache | 160.96 | 49.44 | 2.75 | 0.87 |
| 1500-KB memory cache | 123.19 | 51.44 | 2.65 | 0.68 |
| 3000-KB memory cache | 124.22 | 52.21 | 1.68 | 0.87 |
| 1500-KB memory cache + 10-MB disk cache | 118.71 | 49.63 | 2.73 | 0.88 |
| 1500-KB memory cache + 20-MB disk cache (default) | 118.14 | 47.38 | 2.81 | 0.88 |
| 1500-KB memory cache+ 40-MB disk cache | 119.54 | 48.08 | 1.74 | 0.88 |
| 1500-KB memory cache + 80-MB disk cache | 120.28 | 48.23 | 2.83 | 0.88 |

**Table 4:** Bandwidth consumption per cache size per user scenario

**Chart 4:** Memory Cache (MC) and Disk Cache (DC) sizes versus bandwidth

Overall, when memory or persistent storage on the clients is not an issue, it would be beneficial to enable larger memory caches and persistent storage to improve bandwidth consumption. Depending on the scenario, the results will be different, but as client and server can retrieve more data from their caches, fewer bytes have to be sent or resent over the wire.

## Desktop Themes

A theme is a preset package containing graphical appearance details, and is used to customize the [look and feel](http://en.wikipedia.org/wiki/Look_and_feel) of the desktop and window components. You can enable the Windows Vista theme on Windows Server 2008 to get the look and feel of the Windows Vista desktop. Themes can be enabled on Windows Server 2008 by installing the Desktop Experience pack and enabling the theme service. However, this increased graphics overhead causes an increase in bandwidth consumption.

Table 5 and Chart 5 show the bandwidth consumption (in KBps) for user scenarios at different color depths with and without themes enabled.

|  |  |  |  |
| --- | --- | --- | --- |
| **User scenario** | **Theme on** | **Theme off** | **Performance loss due to themes** |
| Typing and Scrolling (16-bit) | 2.82 | 1.64 | 72.2477772 % |
| Typing and Scrolling (32-bit) | 20.20 | 4.35 | 364.5638076 % |
| Internet Explorer (32-bit) | 90.47 | 12.96 | 597.9317742 % |
| Internet Explorer (16-bit) | 19.71 | 5.91 | 233.2598376 % |

**Table 5:** Bandwidth consumption at different color depths with and without themes enabled

**Chart 5:** Bandwidth consumption at different color depths with and without themes enabled

## Desktop Composition

The new Windows Vista Desktop Composition feature fundamentally changes the way applications display pixels on the screen. When Desktop Composition is enabled, individual windows no longer draw directly to the screen or primary display device as they did in earlier versions of Windows. Instead, their drawing is redirected to off-screen surfaces in video memory, which are then rendered into a desktop image and presented on the display.

Desktop Composition is performed by the Desktop Window Manager (DWM), a new component of Windows Vista. Through Desktop Composition, DWM enables visual effects on the desktop as well as in various features, such as glass window frames, 3-D window transition animations, Windows Flip, and Windows Flip 3D.

Remote Desktop Protocol was extended to support Desktop Composition remoting (or, composed mode RDP). This is achieved by transmitting composition commands from DWM on the server to the RDC client. The client interprets these commands and renders the desktop. The composition commands lead to increased bandwidth consumption. Desktop Composition remoting is only available from Windows Vista to Windows Vista.

Table 6 and Chart 6 compare bandwidth consumption (in KBps) between composed mode RDP and direct mode RDP.

|  |  |  |
| --- | --- | --- |
| **User scenario** | **Composed mode RDP** | **Direct mode RDP** |
| Executive PPT | 126.87 | 119.43 |
| Simple PPT | 73.79 | 56.78 |
| Typing and Scrolling | 3.02 | 20.20 |
| Scrolling | 2.04 | 2.24 |
| Internet Explorer | 25.17 | 90.47 |

**Table 6:** Bandwidth consumption between composed mode RDP and direct mode RDP

**Chart 6:** Bandwidth consumption between composed mode RDP and direct mode RDP

## RemoteApp Programs

Windows Server 2008 supports Terminal Services RemoteApp (TS RemoteApp), which allows users to remote into an application rather than a complete desktop. This feature blurs the line between local and remote applications and enables a new level of transparency for the end user. For RemoteApp programs to work correctly, RDP detects changes in window position and state on the server side and forwards this information to the client. This causes extra bandwidth consumption, but you also save on the bandwidth consumed by the desktop and shell UI components.

In our testing, we found that bandwidth consumption when using RemoteApp programs could increase and decrease, depending on the application and its usage, when compared to using a remote desktop.

Table 7 and Chart 7 compare bandwidth consumption (in KBps) for RemoteApp programs versus a remote desktop at 16-bit color depth over a 56-Kbps modem.

|  |  |  |  |
| --- | --- | --- | --- |
| **User scenario** | **RemoteApp programs** | **Remote desktop** | **% performance gain using RemoteApp programs** |
| Executive PPT | 125.03 | 122.65 | -1.94 |
| Simple PPT | 61.71 | 50.49 | -22.21 |
| Typing and Scrolling | 2.91 | 2.82 | -3.17 |
| Scrolling | 0.79 | 0.88 | 10.06 |

**Table 7:** Bandwidth consumption when using RemoteApp programs versus a remote desktop

**Chart 7:** Bandwidth consumption when using RemoteApp programs versus a remote desktop

# Bulk Compression

Remote Desktop Protocol 6.1 introduces two new bulk compressors that compress all data sent from the server to the client. These are available in addition to the bulk compressor present in RDP 5.0. These compressors can be enforced by the computer-wide **Set compression algorithm for RDP data** Group Policy setting. The choice of compression algorithm impacts the memory and CPU consumption on the server and thus affects server scalability. You need to be cognizant of this before changing the compressor settings. The following is a description of the bulk compressor settings:

1. **Optimized to use less memory (RDP 5.2 or V1):**
   * Bulk compressor from Windows Server 2003
   * Consumes more bandwidth than other compressors
   * Has the least memory and CPU overhead
   * Gives you the best server-side scalability
2. **Balances network bandwidth and memory (RDP 6.0 or V2):**
   * The default setting if the Group Policy setting is not configured
   * Balances between memory consumption and network bandwidth
   * Can reduce your bandwidth by 5–30 percent compared to the RDP 5.2 compressor
3. **Optimized to use less network bandwidth (RDP 6.1 or V3):** 
   * A new compressor designed for Windows Server 2008
   * Tuned to give you the best network performance
   * Can reduce your bandwidth by 10–60 percent compared to the RDP 5.2 compressor

Table 8 and Chart 8 show the bandwidth consumption (in KBps) of user scenarios for different bulk compressor settings. These tests were conducted at 16-bit color depth over a 56-Kbps modem.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User scenario** | **RDP 6.1** | **RDP 6.0** | **RDP 5.2** | **Performance gain: RDP 6.0 vs. RDP 5.2** | **Performance gain: RDP 6.1 vs. RDP 5.2** | **Performance gain: RDP 6.1 vs. RDP 6.0** |
| Executive PPT | 108.83 | 122.65 | 169.49 | 27.64 | 35.79 | 11.27 |
| Simple PPT | 28.96 | 50.49 | 52.97 | 4.68 | 45.34 | 42.65 |
| Typing and Scrolling | 1.76 | 2.82 | 4.80 | 41.24 | 63.42 | 37.74 |
| Scrolling | 0.65 | 0.88 | 1.11 | 20.33 | 41.09 | 26.06 |
| Internet Explorer | 6.58 | 19.71 | 14.60 | -35.01 | 54.90 | 66.60 |

**Table 8:** Bandwidth consumption for different compressor settings using 16-bit color depth

**Chart 8:** Bandwidth consumption for different compressor settings using 16-bit color depth

Table 9 and Chart 9 show the bandwidth consumption (in KBps) of user scenarios for different bulk compressor settings. These tests were conducted at 32-bit color depth over a 56-Kbps modem.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User scenario** | **RDP 6.1** | **RDP 6.0** | **RDP 5.2** | **Performance gain: RDP 6.0 vs RDP 5.2** | **Performance gain: RDP 6.1 vs RDP 5.2** | **Performance gain: RDP 6.1 vs RDP 6.0** |
| Executive PPT | 102.39 | 119.43 | 160.10 | 25.40 | 36.04 | 14.27 |
| Simple PPT | 32.46 | 56.78 | 60.72 | 6.49 | 46.55 | 42.84 |
| Typing and Scrolling | 3.34 | 20.20 | 22.42 | 9.92 | 85.11 | 83.47 |
| Scrolling | 1.02 | 2.24 | 2.53 | 11.35 | 59.60 | 54.42 |
| Internet Explorer | 10.38 | 90.47 | 76.42 | -18.39 | 86.41 | 88.52 |

**Table 9:** Bandwidth consumption for different compressor settings using 32-bit color depth

**Chart 9:** Bandwidth consumption for different compressor settings using 32-bit color depth

Table 10 details the theoretical per session CPU and memory overhead for each bulk compressor.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **RDP 5.2** | **RDP 6.0** | **RDP 6.1** |
| **Average memory consumption per session** | 128 KB | 328 KB | 2.6 MB |
| **Average CPU consumption per session** | 0.5% | 1.5% | 1.0% |

**Table 10:** Theoretical per session CPU and memory overhead for each compressor

# Performance Comparison with Earlier RDP Versions

This section compares the performance of different versions of Remote Desktop Protocol.

In our tests, the data shows that bandwidth consumption for:

* Windows Vista to Windows Server 2008 with the RDP 6.0 compressor is up to 30 percent better than Windows Vista to Windows Server 2003 depending on the scenario.
* Windows Vista to Windows Server 2008 with the RDP 6.1 compressor is up to 45 percent better than Windows Vista to Windows Server 2003.

Table 11 and Chart 11 show the bandwidth consumption (in KBps) between a Remote Desktop Connection client and different server operating systems. The tests were conducted at 16-bit color depth with themes turned off.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **User scenario** | **Windows Server 2003 (RDP 5.2)** | **Windows Server 2008**  **(RDP 6.0)** | **Windows Server 2008**  **(RDP 6.1)** | **% performance gain: 2008 (RDP 6.0) vs 2003 (RDP 5.2)** | **% Performance gain: 2008 (RDP 6.1) vs 2003 (RDP 5.2)** |
| Executive PPT | 164.33 | 117.33 | 97.82 | 28.60 | 40.48 |
| Simple PPT | 60.93 | 51.46 | 33.43 | 15.54 | 45.14 |
| Typing and Scrolling | 1.69 | 1.69 | 1.56 | 0.12 | 7.73 |
| Scrolling | 0.70 | 0.64 | 0.57 | 8.12 | 17.88 |
| Internet Explorer | 7.98 | 6.45 | 5.59 | 19.23 | 29.99 |

**Table 11:** Bandwidth consumption between RDC client and different server operating systems

**Chart 11:** Bandwidth between RDC client and different server operating systems

Comparing the performance of sending graphics data across the different protocol versions by using a benchmark tool such as WinBench also demonstrates performance gains when using RDP 6.1 over RDP 5.2.

Table 12 and Chart 12 show the bytes sent over the wire running tests by using WinBench 99. The tests were conducted between different clients and the same server running Windows Server 2008. This ensures that we are sending the same amount of graphics data across the channel. The tests were conducted at 16-bit color depth with WinBench99 on the server.

|  |  |  |  |
| --- | --- | --- | --- |
| **RDC client version** | **Server-side operating system** | **Bytes** | **Performance gain using RDP 6.1** |
| RDC 6.1 (Windows Server 2008) | Windows Server 2008 | 7559075 |  |
| RDC 5.2 (Windows Server 2003) | Windows Server 2008 | 9450351 | 20% |
| RDC 5.1 (Windows XP) | Windows Server 2008 | 11185633 | 32% |

**Table 12:** Bytes sent over the wire running tests by using WinBench 99

**Chart 12:** Bytes sent over the wire running tests by using WinBench 99 Flow Control

## Bandwidth Allocation for Terminal Server Connections over RDP

Over a terminal server RDP connection, multiple devices (such as video, Clipboard, and printers) send data over the connection from server to client. On a low bandwidth connection these applications compete for available bandwidth. As a result, important graphics data has to compete with data transmitted in the background, such as a print job or file copy. This problem manifests itself most severely when printing a large document over a low bandwidth connection. The printer data competes for available bandwidth with the video rendering causing significant deterioration in the graphics rendering.

In Windows Server 2008 we fix this by introducing a simple scheme wherein a fixed percentage of bandwidth is allocated to video, and the rest goes to virtual channel traffic (all other redirected devices). By default this allocation is 70 percent for graphics data and 30 percent for virtual channel data, meaning when bandwidth usage is under pressure, graphics data is guaranteed to get 70 percent of the available bandwidth.

Although this scheme effectively solves the problem, you might want to tweak the settings a bit for some scenarios. To change the settings, you can set registry values. For these settings to take effect, the computer must be restarted.

Following is the list of registry values that affect the bandwidth allocation behavior. These are all DWORD values under **HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\TermDD**:

* **FlowControlDisable:** When set to 1, this value disables the new flow control algorithm, making it essentially First-In-First-Out (FIFO) for all packet requests. This provides results similar to Windows Server 2003. (The default for this value is 0).
* **FlowControlDisplayBandwidth / FlowControlChannelBandwidth:** These two values together determine the bandwidth distribution between display and virtual channels. You can set these values in the range of 0–255. For example, setting FlowControlDisplayBandwidth = 100 and FlowControlChannelBandwidth = 100 creates an equal bandwidth distribution between video and VCs. The default is 70 for FlowControlDisplayBandwidth and 30 for FlowControlChannelBandwidth, thus making the default distribution equal to 70–30.
* **FlowControlChargePostCompression:** If set to 1, this value bases the bandwidth allocation on post-compression bandwidth usage. The default for this value is 0, which means that the bandwidth distribution is applied on pre-compressed data.

# Device Redirection

## Audio

The audio remoting stack caps the audio signal being transmitted to 22 kilohertz (kHz), resulting in a maximum theoretical bandwidth of 86 KBps. The RDP stack then compresses the audio signal further by using in-box audio codecs. The choice of the codec is determined by codecs supported by the client and server, as well as the bandwidth available between the client and the server.

Table 13 shows the results of our internal tests which involve playing in-box Windows Vista audio samples by using Windows Media® Player over a LAN connection.

|  |  |  |
| --- | --- | --- |
| **Sample name** | **Length(sec)** | **KBps** |
| Windows startup.wav | 4 | 32.44 |
| Distance.wma | 327 | 21.69862 |
| Windows shutdown.wav | 3 | 29.89333 |

**Table 13:** Playing audio samples by using Windows Media Player over a LAN

## Printer

Terminal Services Easy Print is a proxy for every print action that simply redirects all printing-related work to the user's local computer without the need to install any print drivers on the server that is running Terminal Services. This system provides several benefits, such as being able to redirect any printer from the user's client computer without having to reconfigure the server while still allowing the user to configure the print job as though he were printing on his client computer.

On the performance side, one of the key benefits is in multiple copies of the same print job – legacy Terminal Services printing would cause the bandwidth usage to go up by a factor of *n* if there were *n* print copies of the same document requested on the same redirected printer, whereas Easy Print just sets a flag for the number of copies in the XPS format file that is transmitted over the wire, resulting in clear bandwidth savings.

# Conclusion

The test results, and more importantly, the actual end user experience, demonstrate that the performance of Remote Desktop Protocol 6.1 in Windows Server 2008 and Windows Vista is substantially better than earlier RDP versions, resulting in an improved user experience and lower network bandwidth usage.