Microsoft RemoteFX for Session Virtualization: Architectural Overview

Abstract

Windows Server® 2008 R2 Remote Desktop Services (Remote Desktop Services) includes a new set of user experience technologies in Service Pack 1 (SP1) called Microsoft® RemoteFX™.

RemoteFX adds new capabilities to the Remote Desktop Services platform by delivering a full-fidelity Virtual Desktop Infrastructure (VDI) and high-density, scalable, client-agnostic session virtualization capabilities, shifting delivery intelligence to host-based systems running RemoteFX. Through a new graphics payload purpose-built for RemoteFX and Remote Desktop Protocol (RDP), the session virtualization platform is tightly integrated with the RDP protocol, which enables shared encryption, authentication, management, and local device support.

This paper provides an architectural overview of RemoteFX in the context of session virtualization—using a new Windows Server role called the Remote Desktop Session Host (RD Session Host) designed specifically for modern session virtualization environments using Windows Server 2008 R2 SP1.
Disclaimer

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.

©2011 Microsoft Corporation. All rights reserved.

Microsoft, Access, Active Directory, Aero, DirectX, Excel, Hyper-V, Powerpoint, PowerShell, RemoteApp, RemoteFX, Silverlight, SharePoint, Windows, Windows Server, and Windows Vista 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.
# Table of Contents

RemoteFX Overview....................................................................................................................... 3

From Terminal Services to Session Virtualization........................................................................... 3
  
  Building on Client-centric Computing........................................................................................... 4
  Transitioning to Host-centric Session Virtualization..................................................................... 4

RemoteFX for Remote Desktop Session Host ................................................................................... 5
  
  What’s New in RDP in 7.1 ............................................................................................................. 5
  New Concepts for RDP 7.1 ............................................................................................................. 6

RemoteFX Platform Architecture for RD Session Host ................................................................. 6

Terminal Server Core and RDP Integration....................................................................................... 7

RemoteFX Architectural Concepts ................................................................................................ 8

RemoteFX for Session Virtualization Pipeline ............................................................................... 9

RDP 7.1 Client and Graphics Rendering Pipeline .......................................................................... 10

Encoding and Decoding Options .................................................................................................. 11

RemoteFX Architectural Benefits ................................................................................................. 11

Session Virtualization Deployment Considerations ...................................................................... 12

User Scenarios ............................................................................................................................... 12

Application Fit .............................................................................................................................. 12

Visually Rich Applications .......................................................................................................... 12

Performance ................................................................................................................................. 12

RemoteFX Ecosystem Partners .................................................................................................... 13

System Requirements .................................................................................................................. 13

Hardware Requirements .............................................................................................................. 13

Software Requirements ............................................................................................................... 13

Summary ....................................................................................................................................... 13
RemoteFX Overview

A breakthrough in the way knowledge workers interact with rich 2D and 3D applications is underway in the data center. Instead of relying purely on local computing power, a new graphics and session virtualization platform from Microsoft is shifting graphics-processing intelligence to centrally-hosted virtual desktop and session virtualization infrastructures deployed in enterprise data centers.

Microsoft RemoteFX is a platform technology in Windows Server 2008 R2 Service Pack 1 that introduces a new set of user experience enhancements to remote desktop computing that enables a rich, local-like desktop environment viewed over the network. As an integral part of Remote Desktop Services, RemoteFX ensures continuity with existing Remote Desktop Protocol (RDP) features while offering new virtualization capabilities for Virtual Desktop Infrastructure (VDI) and session virtualization (formerly known as Terminal Services).

For session virtualization, this new set of capabilities provides increased session performance and compatibility across an ecosystem of client-side devices. These devices range from new ultra-lightweight and low-powered thin clients to new form factors such as LCD displays (with integrated RemoteFX support) to rich clients such as traditional computer workstations.

For virtual desktops that are deployed on the Hyper-V virtualization platform, an architectural overview on RemoteFX for VDI, using the Remote Desktop Virtual Host (RD Virtualization Host) role in Windows Server 2008 R2 SP1, is provided in the following paper:

Microsoft RemoteFX for Virtual Desktop Infrastructure: Architectural Overview

Through a new but familiar system role, the Remote Desktop Session Host (RD Session Host), RemoteFX capabilities are made available to IT professionals familiar with managing Terminal Services.

From Terminal Services to Session Virtualization

Since its inception in 1997, Microsoft Terminal Services (now Remote Desktop Services) has evolved into a proven session virtualization platform. Over the years, continuous improvements in performance, scalability, reliability, RDP, and other enhancements, have given IT professionals a mature platform for consolidating targeted applications in support of centralized remoting initiatives across the enterprise.

Today, customers need to further increase performance, support new classes of low power devices, and improve the end user visual experience. Moreover, customers want to maintain familiar tools and methodologies learned and deployed over the years using Terminal Services, and ensure new technologies support deployed security infrastructures and policies.

RemoteFX provides higher performance and new core service optimizations for session virtualization. The RemoteFX platform is graphics- and device-agnostic, enabling an actively expanding ecosystem of client devices, graphics types, and access scenarios. These scenarios range from classic terminal services
scenarios for hosting low-usage legacy applications to new hosted desktop scenarios for knowledge and task workers using new line-of-business applications well suited to a session-based architecture.

**Building on Client-centric Computing**

Microsoft believes session virtualization should complement and build on investments already made by IT, including the deployed physical client infrastructure, applications, and user environments. Remoting solutions should maintain familiarity for end-users and administrators by providing greater flexibility and improving the user experience (versus degrading or complicating it).

Client devices should be easier to deploy, manage, and use. Customers expect users to plug-and-play IT-approved peripheral devices, and easily connect new devices to a user’s primary device. New devices need to *just work* within a session virtualization or VDI environment.

Enterprise customers need to increase utilization of deployed devices. Those rich clients (from legacy computers to high-end graphics workstations) should complement host-based remoting techniques.

While traditional terminal services remoting seems *server-centric* (much processing occurs on the host), historically, devices such as personal computers and thin clients needed to be relatively powerful in order to enable a rich experience for the local user. Capabilities on client devices needed to closely match capabilities on the server host. When a session is launched on the host, graphics information must be efficiently processed and sent to the client device, in a client-dependent stream. If the client device is powerful, contains the right software, and maintains corresponding graphics intercept points with the host, a high-quality user experience over relatively low-bandwidth can be achieved.

If a less powerful client device such as a low-power thin client is used, this strategy could degrade the user experience (through choppy video, delayed audio, or missing graphics). In these cases, processing should be performed on the host and optimally streamed to the client.

**Transitioning to Host-centric Session Virtualization**

A consistent user experience combining the benefits of client-side computing with host-side computing can be achieved for session virtualization—supporting a broad number of devices and form-factors—by shifting the intelligence and raw computing power away from the client and into the data center.

Today, bandwidth is less expensive and broadly available. Additionally, a wide array of graphics formats (for example, HTML5, Silverlight, Adobe Flash, DirectX, Aero Glass, Windows Media) are relevant to Windows users. These changing conditions require a new hosting model that can broadly support common graphics types across a broader array of diverse client endpoints; some powerful; some not.

With host-centric remoting, all the graphics can be intercepted on the host at an optimal layer in the software stack. All graphics are rendered on the host into a single frame buffer (a temporary holding station for graphical updates) that represents the end-user display. Changes to the frame buffer are sent to the client at a frame rate that dynamically adapts to network conditions and the client’s ability to consume the changes.
Through a new graphics payload purpose-built for RDP, combined with host-side session processing optimizations, clients can be of nearly any level of complexity and form factor. Many graphics types can be supported in a common way through low-level intercept points enabled on the host. Fewer dependencies on the client device improve the user experience and expand compatibility across a broader range of device types.

**RemoteFX for Remote Desktop Session Host**

Remote Desktop Services provides a set of technologies, including RemoteFX, to enable users to access session-based desktops and hosted applications through RDP using the Remote Desktop Session Host (RD Session Host) role.

In Windows Server 2008 R2, the RD Session Host role was enhanced with:

- Higher native session performance and scalability. New kernel-mode components increase CPU distribution for hosted sessions, per-user.
- Session fairness: an algorithm fairly distributes CPU cycles across sessions. The algorithm dynamically distributes cycles based on the number of active sessions and their load.
- Scale-up performance: Processor resources are shared on a per CPU basis enabling multi-socket and -core scenarios.

In Windows Server 2008 R2 SP1, RemoteFX adds the following improvements:

- New RDP Payload: a new graphics payload is delivered with RDP 7.1 providing a device-agnostic, host-optimized delivery format for Windows and non-Windows devices supporting RDP 7.1 or later (described below in more detail).
- RemoteFX Encode Library: A new encoding library integrates with the RDP Display Driver. This takes output from hosted applications running in user sessions (sessions 1, 2, 3, and so on) optimizing each session through a client-agnostic bitmap conversion.

**What’s New in RDP in 7.1**

While much of RDP has not changed, a new graphics payload is a key innovation in RDP 7.1. This payload integrates familiar and new RDP capabilities with an extensible decoder designed for RemoteFX client devices. On the host, RemoteFX provides a single, low-level graphics intercept point for host-side processing. The resulting architecture supports real-time rendering, compression, and decoding— projected across the network using a diverse ecosystem of RemoteFX-ready device types.

This new host-integrated payload is critical in achieving client device-agnostic user experiences, across a variety of content types and formats. As new media-intensive standards emerge, such as HTML 5, host-based acceleration using RemoteFX will vastly improve the experience and enable broader adoption of these new formatting standards. For example, in RDP 7.1, visually rich audio and video playback can be experienced in real-time over the network on a broader range of device-types, including new ultra-lightweight thin clients enabled by low-power chip designs embedded in a variety of form factors.
**New Concepts for RDP 7.1**

<table>
<thead>
<tr>
<th>RDP Client-side Rendering</th>
<th>RDP Host-side Rendering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Separate intercept points per graphic stack and equivalents were often prone to gaps in user experiences</td>
<td>Single intercept point for all graphics processing—offers consistent, predictable, and complete user experiences</td>
</tr>
<tr>
<td>Relies purely on software and hardware capabilities of rich client computers and workstations</td>
<td>Supports diverse thin client base by shifting workload and complexity to host systems</td>
</tr>
<tr>
<td>Conserves bandwidth through local processing of intercepted graphics types</td>
<td>Although optimized for efficient delivery, typically requires higher bandwidth for sending compressed bitmaps and other formats over the wire</td>
</tr>
</tbody>
</table>

**RemoteFX Platform Architecture for RD Session Host**

RemoteFX for RD Session Host is a platform architecture built on Remote Desktop Services optimized for session virtualization.

For session virtualization, RemoteFX is designed from the ground up to achieve high-density concurrent session hosting for workloads that are less visually demanding than VDI. In most scenarios for session virtualization it is sufficient to support 2D rendering capabilities. The following section provides an overview of the RemoteFX architecture.

RemoteFX incorporates kernel- and user-mode components with new encoding techniques to increase session performance, concurrency, and density. Using host-based encoding techniques performance can be further increased. Through integration with the new graphics payload designed for RDP, a broader range of graphics and client device types are supported.

A variety of partner-based solutions for VDI and session virtualization are purpose-built for the RemoteFX platform to include advanced encoding and decoding techniques in both software and hardware. These solutions provide an array of client device options for RemoteFX.
Terminal Server Core and RDP Integration

RemoteFX for RD Session Host and its associated components for RDP are integrated with TS Core and RDP Core, technologies now included with the RD Session Host role; by contrast, RemoteFX for VDI using the RD Virtualization Host role is built on Hyper-V’s type-1 hypervisor architecture.

RemoteFX for RD Session Host uses the Terminal Server-derived concept of session 0 and session 1 and greater. Session 0 contains isolated system processes used by core system components. Sessions 1 and greater represent individual user or RDP sessions.

RemoteFX for RD Session Host also uses the Windows Server concept of user and kernel mode. Sessions for TS Core run in user-mode of Windows Server 2008 R2 SP1. TS Core includes LSM, TERMSRV, and RDPWSX. User sessions and individual user applications also run in user mode. RemoteFX management components for session virtualization, display management, and encoding are installed in kernel mode.
# RemoteFX Architectural Concepts

<table>
<thead>
<tr>
<th>Enabling Technology</th>
<th>Customer Value</th>
<th>Differentiating Innovation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtual GPU (RD Virtualization Host only)</td>
<td>Full rich Windows experience</td>
<td>Content- and hardware-independent intercepts and rendering</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Single or multiple shared GPU(s) for multiple virtual machines</td>
</tr>
<tr>
<td>Host-side Rendering (RD Session Host and RD Virtualization Host)</td>
<td>Remoting of any content</td>
<td>Applications run at full speed on host</td>
</tr>
<tr>
<td>Intelligent screen capture and hardware-based encode (RD Session Host and RD Virtualization Host)</td>
<td>High-fidelity user experience</td>
<td>Screen deltas sent to clients based on optimization policies and client and network availability</td>
</tr>
<tr>
<td>Bitmap remoting and hardware-based decode (RD Session Host and RD Virtualization Host)</td>
<td>Broad range of client device support</td>
<td>CODEC designed for text and image content</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Single CODEC for VDI and session virtualization scenarios</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Choice of software and hardware decoding options</td>
</tr>
</tbody>
</table>
**RemoteFX for Session Virtualization Pipeline**

When a user launches an RDP client session, a new session (1 or greater) is launched on the server. Within each session, as applications are launched, graphics commands are made through graphics device interface (GDI), in user mode. GDI makes calls into Win32K.sys, the kernel-mode counterpart to GDI.

![RemoteFX for Remote Desktop Session Host Pipeline](image)

For each session, Win32K.sys loads a separate instance of a 2D display driver called RDPDD (Remote Desktop Protocol Display Driver). Each session spawns a separate instance of RDPDD. (This is analogous to a virtual machine used in a VDI scenario, running an instance of the vGPU.)

RemoteFX encoding is a new capability included in RD Session Host with SP1. This capability delivers RDP-optimized bitmap creation for delivery across RDP. Effectively, the output streams from session-hosted applications are encoded using the internal CPUs of the RD Session Host system.
Once a stream is encoded, the RDP Display Driver outputs the stream to RDP Core, running in kernel mode. From here, the RDP Core stack delivers RDP session to RDP clients over the wire. This delivery process across RDP Core is similar to the RD Session Host system delivered in R2. The listening socket for RDP 7.1 is the same as RDP 7. From a network standpoint, the RDP 7.1 stack in User Mode looks identical to the RDP 7 stack used in the previous version of RDP.

However, the end-to-end host-to-client delivery pipeline across RDP is new.

**RDP 7.1 Client and Graphics Rendering Pipeline**

Once an encoded payload is ready to be output, RDP streams the payload across virtual channels, a form of connection within RDP. Virtual channels are software extensions that can be used to extend Remote Desktop Services (for example, support for special types of hardware and audio). The RDP protocol provides multiplexed management of multiple virtual channels. A virtual channel application has two parts, a client module and a server module. The server module is an executable program running on the Remote Desktop Session Host (RD Session Host) server. The client module is a DLL that must be loaded into memory on the client computer when the Remote Desktop Connection (RDC) client program runs.

RemoteFX uses a special graphics virtual channel to distribute screen information to clients (and other channels for audio, device input, etc.). Through this virtual channel in RDP, data is sent to client-side RemoteFX decoding abstraction layer where data is decompressed. From this layer, there are two main scenarios for delivering streams via RDP to clients:

1. Classic RDP (Non-RemoteFX using RDP 7 or earlier). In this case, the client is responsible for rendering data, based on the content-specific formatting, using graphics providers (such as for GDI,
Aero, or Windows Media) loaded by the RDP client. The benefit is low network traffic, but requires client devices capable of rendering graphics content. These devices are typically more complex and expensive.

2. RemoteFX Host-Optimized Decoding: In cases where the content was delivered from RemoteFX using RDP 7.1 or later, data is streamed across virtual channels into a RemoteFX decoding abstraction layer. Based on the target device capability, content is either decoded in software using client-side GPU or CPU resources, or directly in client-side hardware, through a dedicated decoder. These clients can range from fully functional to simple, low-cost device types.

**Encoding and Decoding Options**

Through software, RemoteFX on RD Session Host utilizes industry-standard CPUs for encoding. Hardware-offload encoding techniques enabled through partner solutions may also be available in the future.

The basic decoding requirement for the client end-point is supporting the ability to decode and display highly compressed bitmaps. At a minimum, the client needs the decoder counterpart to the encoder that was used on the host as well as very basic graphics display capability.

Payloads over RDP can be decoded through software or hardware, based on the client device-type and its built-in capabilities. If data streams are decoded using software, the CPU of the target device is used. If data streams are decoded using hardware, this can be achieved using the CPU or GPU of the target, or combination of the two. If an ASIC or another embedded design is available, decoding can be fully offloaded from the target client device.

**RemoteFX Architectural Benefits**

There are a number of benefits to this host-based architecture:

- Network agnostic: Any standard IP network can be used across the same established RDP security ports and policies. Networking security remains consistent with RDP.
- Client agnostic: RDP 7.1 supports a wide range of clients with various hardware and software decoding options.
- Payload-optimized: RemoteFX provisions and transports only relevant delta changes through an innovative differencing algorithm. This is similar to network-based deduplication and compression techniques.
- Frame-rate optimized: Payloads can be controlled through Group Policy in Active Directory and based on network conditions, such as availability.
- Device-transparent: The decoding process is transparent to the user and device. While the output to the client depends on client type, the payload decision is processed on the server. For example, a regular computer might receive one format while a light thin client would receive a bitmap.
Session Virtualization Deployment Considerations

User Scenarios
Common user types that benefit from Remote Desktop Services are task workers (such as factory floor and call center workers), contractor/offshore workers, knowledge workers, and remote workers in branch offices.

Application Fit
Session virtualization is well suited to task worker or simple knowledge worker scenarios where the primary need is for text-based and graphical 2D applications. These range from legacy applications to vertical line-of-business applications to some rich media applications. Other examples include: applications designed to support regulatory and compliance initiatives, complex applications, line-of-business applications in support of specific verticals or business functions, and scenarios where large and complex applications are frequently updated.

Visually Rich Applications
New or legacy visual 2D applications requiring an improved user experience are well suited to RemoteFX for RD Session Host. These include applications written for GDI. For session virtualization, RemoteFX supports a range of portable graphics content used by visually rich 2D applications.

Performance
Performance between the RD Session Host system and individual client device depends on a number of factors. Detailed performance guidelines are outside the scope of this paper.

Generally, performance factors include but are not limited to:

- Host CPU capabilities: The scale of server, CPU, and number of sockets and cores. RemoteFX for RD Session Host supports multiple CPUs on the host and is designed for scaling in the enterprise.
- Host hardware encoding capabilities: Future embedded ASICs may increase encoding performance and overall session density per host.
- Client device capabilities: Performance can be increased by utilizing dedicated hardware-based decoders and increasing client CPU.
- Network capabilities: More bandwidth can be utilized to enable higher performance.

Fair Share CPU Scheduling
Fair Share CPU Scheduling is a feature included with RD Session Host to dynamically distribute processor resources across sessions based on the number of active sessions and load on those sessions by using the kernel-level scheduling mechanism included with Windows Server 2008 R2. On an RD Session Host server, one user will not diminish the performance of another user’s session under load.
Network Characteristics
While the RemoteFX platform is currently optimized for LAN environments, Microsoft is working closely with partner companies on wide-area optimization solutions to enable high fidelity user experiences across higher latency and lower bandwidth WANs.

RemoteFX Ecosystem Partners
Microsoft has a large partner ecosystem building solutions on the Remote Desktop Services platform, including RemoteFX.

For RemoteFX, solution partners provide both software and hardware solutions. Hardware solutions range from host-side GPUs and ASICs to embedded client-device "system on chip" architectures where a chip has multiple capabilities. Software solutions expand the range of server, client, and network configurations for which RemoteFX can be successfully deployed.

System Requirements

Hardware Requirements
To take advantage of RemoteFX functionality when connecting to RD Session Host, the server must meet the hardware requirements listed in the Windows Server Technical Library.

For more information about the hardware requirements, see Hardware Considerations for RemoteFX.

Software Requirements
RD Session Host Physical System
Windows Server® 2008 R2 SP1 Remote Desktop Services Virtualization Host (RD Session Host).

RDP Client
Host-optimized RDP clients using RemoteFX: any device supporting RDP 7.1 or later.

Summary
Microsoft RemoteFX introduces a new set of user experience enhancements to remote desktop computing in Windows Server 2008 R2 SP1 that enable a rich, local-like desktop environment over the network. With RemoteFX, connected knowledge and task workers can experience their workspace in full fidelity, improving user productivity.

RemoteFX shifts certain session processing functions to centrally-hosted session virtualization hosts to increase performance, support a broader range of client devices, and complement and extend user
experience enhancements provided in Remote Desktop Services. These new capabilities provide increased performance at higher densities, and an improved user experience projected across a broader range of client devices.

These new technologies are compatible with deployed terminal services and session hosting security infrastructures. Applications and data remain secure and centrally managed. The management tools and technologies provide an easy transition for IT professionals familiar with managing Terminal Services and other Microsoft technologies.

Microsoft RemoteFX, supported by a growing partner ecosystem, provides the cornerstone for advancing graphics and session virtualization in support of a wide range of contemporary and new media rich remoting scenarios, across the enterprise and into data center cloud hosting initiatives.