

Multi-Codec Graphics Pipeline for RDP on Linux

Adaptive Content Encoding via the EGFX Protocol

Author: Greg Lamberson, [Lamco](#) Development

Date: March 2026

Web: lamco.ai

Introduction

This document examines the RDP Graphics Pipeline Extension (MS-RDPEGFX) as it was designed: a multi-codec graphics system. The EGFX wire format carries per-PDU codec identifiers. The framing model supports mixed-codec frames. The capability exchange negotiates codec availability per session. Microsoft's Azure Virtual Desktop uses this multi-codec model in production, encoding text with ClearCodec, video with H.264, and photographic content with progressive refinement — all within single frame boundaries.

Most open-source RDP implementations treat EGFX as an H.264 delivery mechanism. This works, but it leaves most of the protocol's design unused. ClearCodec — the lossless text codec that the spec declares mandatory for all EGFX implementations — is not implemented in any open-source RDP server. Since approximately 80% of typical remote session content is text, this means every open-source implementation is either applying lossy video compression to its dominant content type or falling back to raw bitmap.

On Linux, the transition from X11 to Wayland creates both challenges and opportunities. The Wayland compositor's damage tracking, surface metadata, and content type signals provide information that maps directly to codec selection decisions — enabling content-aware encoding without pixel analysis.

1. The Three Eras of RDP Graphics

Era 1: Raw Bitmap (1998–2009). BitmapUpdateData PDUs. Raw pixel data. A single 1920×1080 frame at 32bpp is 8MB uncompressed.

Era 2: RemoteFX Codec (2010–2019). Wavelet-based compression tied to RemoteFX vGPU. Disabled in 2020 due to CVE-2020-1036. The codec remains in the spec but Microsoft has invested nothing in it since.

Era 3: Graphics Pipeline Extension (2012–present). 18 major protocol revisions. Azure Virtual Desktop, Windows 365, and all Microsoft cloud desktop services use EGFX exclusively. The trajectory is unambiguous.

2. The Multi-Codec Model

EGFX supports **eight codec identifiers** per PDU:

Codec	Type	Purpose
ClearCodec	Lossless	Text, UI, icons (~80% of content)
AVC420/AVC444	H.264	Video, motion content
Progressive	Wavelet	Photos, gradients
RemoteFX	Wavelet	Natural images
Uncompressed	Raw	Baseline fallback
Planar	Simple	Simple graphics

Within a single frame, different tiles use different codecs. The client decodes each according to its codec ID. The server selects per-tile, per-frame.

Azure Virtual Desktop Production Practice

"Graphics data is separated depending on its content. Text, images, and videos are encoded using a mix of codecs." — Microsoft Azure VDI Documentation

3. ClearCodec: The Mandatory Codec Nobody

Implements

The spec is unambiguous: *"Implementers have to support the ClearCodec codec."*

ClearCodec uses three layers: Residual (RLE for flat colors), Bands (V-Bar dictionary cache — text characters decomposed into vertical pixel columns that repeat across frames), and Subcodec (NSCodec/RLEX for complex regions). After cache warming, V-Bar hit rates exceed 80% for text content.

Despite being mandatory, ClearCodec encoding exists in **no open-source RDP server**.

4. Linux-Specific Advantages

The multi-codec model requires content classification. On Windows, DWM provides this intrinsically. On Linux, the Wayland compositor provides equivalent signals without pixel analysis:

Signal	Source	Classification
<code>wp_content_type_v1</code>	Application declaration	photo, video, game, none
Damage frequency	Compositor	30+ Hz = video, 0 Hz = static
Buffer type	Wayland	SHM = text/UI, DMA-BUF = GPU content
Application ID	<code>xdg_toplevel</code>	Known app classification

A Linux RDP server has content classification that is more reliable, lower-cost, and lower-latency than pixel analysis.

5. The Capture Pipeline: Where Timing Matters

Multi-codec encoding requires timing awareness. Empirical measurement across compositors:

Compositor	Frames/90s	FPS	p50	Range
Sway 1.11	315	3.5	164ms	6ms – 1,033ms
Hyprland 0.54.2	1,953	21.7	44ms	19ms – 148ms

Same protocol, both wlroots-based — 6× difference in frame delivery. The client cannot determine why. This gap limits adaptive encoding: without knowing capture cost, the encoder must assume worst case.

6. Health Monitoring and Observability

A multi-codec pipeline produces richer telemetry. Per-codec encoder statistics enable closed-loop adaptation:



Session health tracking (video, input, clipboard, session lifecycle), adaptive frame rate control (5–60 FPS based on activity), latency governance (interactive/balanced/quality modes), and priority-based event multiplexing ensure that graphics congestion never affects input responsiveness.

7. Current State of Open-Source RDP

Capability	Status
EGFX DVC channel	Implemented
H.264 AVC420/AVC444	Implemented
ClearCodec encoding	Not implemented
Progressive encoding	Not implemented
Mixed-frame encoding	Not implemented
Content classification	Not implemented
Capture timing feedback	Not available

Lamco Development has implemented ClearCodec, Progressive, mixed-frame encoding, and Linux-native content classification. These implementations are operational in lamco-rdp-server.

8. Summary

The EGFX protocol defines a multi-codec graphics architecture that Microsoft uses in production. The combination of EGFX's per-PDU codec selection, Wayland's content metadata, health monitoring, and capture pipeline timing creates a natural pipeline for adaptive remote desktop on Linux that is architecturally superior to what a monolithic approach can achieve.

ClearCodec is ~2,500–3,000 lines. Progressive is ~3,000–3,500 lines. These are well-specified protocol mechanisms, not open research problems. The scope is tractable and the benefit is immediate.

Lamco Development · lamco.ai · March 2026

Greg Lamberson · office@lamco.io