UNLOCKING THE FULL POTENTIAL OF APPLE REMOTE DESKTOP WITH REMOTE DESKTOP MANAGER

Manage macOS devices seamlessly within Remote Desktop Manager

Presented by Marc-André Moreau (CTO)
and Marc Beauséjour (SME)

KEY FEATURES OF ARD INTEGRATION

- High Performance & Clear Visuals
- Cross-Platform Support (Windows, macOS, Linux, Android, iOS, Web)
- Cost-Effective: Available in RDM Free and Team editions

BENEFITS OF USING ARD INEGRATION

- Superior Performance with MVS adaptive codec
- Flexible Web-Based Access via Devolutions Gateway
- No additional cost compared to Apple's paid ARD client

GETTING STARTED WITH ARD INTEGRATION

- Explore ARD integration today in RDM Free & Team editions

- Visit Devolutions' Integration Center for more details:

https://devolutions.net/integration-center/apple-remote-desktop/

APPLE ADAPTIVE CODEC (MVS)

Progressive JPEG with a twist

- First pass is always sent quickly
- Second pass is sometimes skipped
- High frame rate = no second pass

Special tile encodings

- black & white, bicolor, DCT (JPEG)
- Tile reuse within same update
- Tile caching between updates

ZLIB CODEC

- Zlib uses the DEFLATE algorithm, like the zip file format

  Think of it like zipping uncompressed RGB pixel data

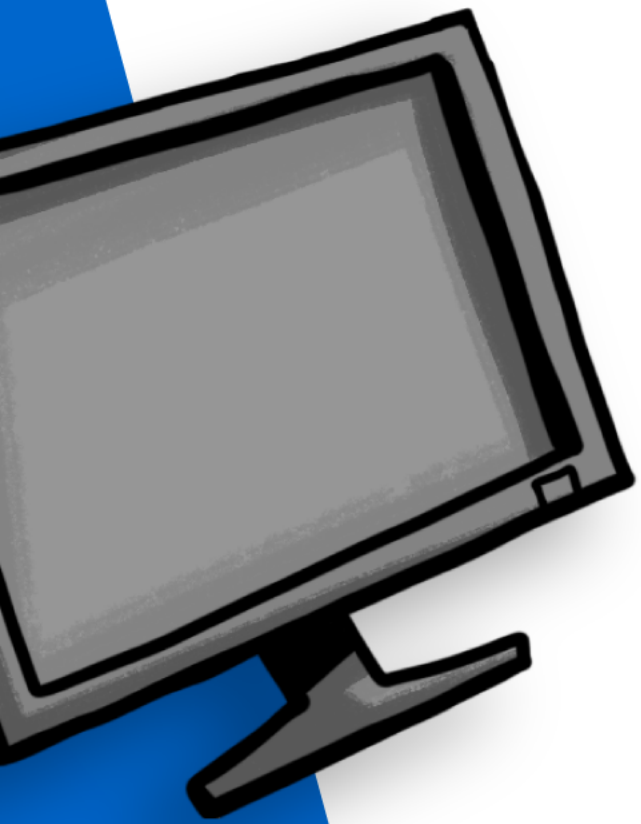
- Supported color formats: 1bpp, 8bpp, 16bpp, 32bpp

  No YUV color transform, no chroma subsampling

- Server-side downscaling supported with ARD

  With standard VNC, you get the full pixel resolution

SERVER-SIDE DOWNSCALING



- Perceived resolution != Native resolution
  - A higher density of pixels is used for one “pixel”
  - Great for local usage, bad for remote desktop
- 1536x960 = 1,474,560 “pixels” (perceived)
- 2048x1280 = 2,621,440 pixels (native)
- High density retina displays use LOTS of pixels
  - This is why server-side downscaling is so important
  - Standard VNC: zlib with no server-side downscaling

PERCEIVED RESOLUTION PIXEL SIZE

Native Resolution	Perceived Resolution	Perceived Resolution Size	Perceived Resolution Gain
2048x1280 = 2,621,440	1792x1120 = 2,007,040	76.5% of native pixel resolution	23.5% fewer pixels to encode
2048x1280 2,621,440	1536x960 = 1,474,560	56.2% of native pixel resolution	43.8% fewer pixels to encode
2048x1280 2,621,440	1344x840 = 1,128,960	43.1% of native pixel resolution	56.9% fewer pixels to encode

COLOR FORMATS

- XRGB: 32 bits per pixel (4 bytes)
  - Alpha is ignored, but takes a whole byte anyway
- RGB: 16 bits per pixel (2 bytes)
  - Visually okay, uses 5 bits per color channel
  - Last bit is simply ignored
- YCbCr (YUV)
  - Y (Luma) uses 8 bits (1 byte) per pixel
  - Cb+Cr (Chroma) uses 16 bits (2 bytes) per group of 4 pixels
  - Modern lossy compression always uses a form of YUV

COLOR FORMAT PIXEL SIZE

Color Format	Size of 4 pixels	Size of 1 pixel	Comment
XRGB (32bpp)	16 bytes (4x4)	4 bytes (8 bits per channel)	Alpha is ignored but takes one whole byte
RGB (16bpp)	8 bytes (4x2)	2 bytes (5 bits per channel)	Last bit is ignored
YCbCr (YUV)	6 bytes (4+2)	3 bytes (8 bits per channel)	Chroma subsampling reduces size of 4 pixels

DEMO

QUESTIONS?

Devolutions

THANK YOU!

https://devolutions.net
https://bsky.app/profile/devolutions.net

https://awakecoding.com
https://bsky.app/profile/awakecoding.com