[MS-RDPEGFX]: Remote Desktop Protocol: Graphics Pipeline Extension

This topic lists the Errata found in [MS-RDPEGFX] since it was last published. Since this topic is updated frequently, we recommend that you subscribe to these RSS or Atom feeds to receive update notifications.

Errata are subject to the same terms as the Open Specifications documentation referenced.

RSS
Atom

Errata below are for Protocol Document Version V13.0 – 2018/03/16.

Errata Published*	Description
2018/07/16	In the following sections, "graphics output buffer" was changed to "Graphics Output Buffer ADM element". Section 2.2.2.14, RDPGFX_RESET_GRAPHICS_PDU Section 2.2.2.15, RDPGFX_MAP_SURFACE_TO_OUTPUT_PDU Section 2.2.2.22, RDPGFX_MAP_SURFACE_TO_SCALED_OUTPUT_PDU For example, in Section 2.2.2.22, RDPGFX_MAP_SURFACE_TO_SCALED_OUTPUT_PDU, changed from: The RDPGFX_MAP_SURFACE_TO_SCALED_OUTPUT_PDU message is sent by the server to instruct the client to map a surface to a rectangular area of the graphics output buffer, including a target width and height to which the surface MUST be scaled. Changed to: The RDPGFX_MAP_SURFACE_TO_SCALED_OUTPUT_PDU message is sent by the server to instruct the client to map a surface to a rectangular area of the Graphics Output Buffer (section 3.3.1.7) ADM element, including a target width and height to which the surface MUST be scaled In that same section, changed from: outputOriginX (4 bytes): A 32-bit unsigned integer that specifies the x-coordinate of the point, relative to the origin of the Graphics Output Buffer (section 3.3.1.7), at which to map the top-left corner of the surface. outputOriginY (4 bytes): A 32-bit unsigned integer that specifies the y-coordinate of the point, relative to the origin of the Graphics Output Buffer, at which to map the upper-left corner of the surface. targetWidth (4 bytes): A 32-bit unsigned integer that specifies the width of the target graphics output buffer to which the surface will be mapped, as specified in section 3.3.1.7. targetHeight (4 bytes): A 32-bit unsigned integer that specifies the height of the target graphics output buffer to which the surface will be mapped. Changed to: outputOriginX (4 bytes): A 32-bit unsigned integer that specifies the x-coordinate of the point, relative to the origin of the Graphics Output Buffer ADM element, at which to map the top-left corner of the surface. outputOriginY (4 bytes): A 32-bit unsigned integer that specifies the y-coordinate of the point, relative to the origin of the Graphics Output Buffer ADM element, at which to map the upper-left corner of the surface.

Errata Published*	Description
	targetWidth (4 bytes): A 32-bit unsigned integer that specifies the width of the target Graphics Output Buffer ADM element to which the surface will be mapped, as specified in section 3.3.1.7. targetHeight (4 bytes): A 32-bit unsigned integer that specifies the height of the target Graphics Output Buffer ADM element to which the surface will be mapped.
2018/06/04	Two sections have been updated to clarify how AVC444/AVC444v2 is encoded and decoded. In Section 2.2.4.5, RFX_AVC444_BITMAP_STREAM, changed from: These bitstreams MUST be decoded by the same MPEG-4 AVC/H.264 decoder as one stream. Changed to: These bitstreams MUST be encoded using the same MPEG-4 AVC/H.264 encoder and decoded by a single MPEG-4 AVC/H.264 decoder as one stream. - Changed from: If no luma frame is present, then this field MUST be set to zero. Changed to: If no YUV420 frame is present, then this field MUST be set to zero. Changed from: 0x1 A YUV420 frame is contained in the avc420EncodedBitstream1 field, and no data is present in the avc420EncodedBitstream2 field. No Chroma420 frame is present. 0x2 A Chroma420 frame is contained in the avc420EncodedBitstream1 field, and no data is present in the avc420EncodedBitstream2 field. No YUV420 frame is present. Changed to: 0x1 A YUV420 frame is contained in the avc420EncodedBitstream1 field, and no data is present in the avc420EncodedBitstream2 field. No Chroma420 frame is present. The Chroma420 frame corresponding to the updates in the YUV420 frame is sent in a RFX_AVC444_BITMAP_STREAM message in subsequent frames if required. 0x2 A Chroma420 frame is contained in the avc420EncodedBitstream1 field, and no data is present in the avc420EncodedBitstream2 field. No YUV420 frame is present. The Chroma420 frame MUST be combined with the decoded AVC stream from previous frames. In Section 2.2.4.6, RFX_AVC444V2_BITMAP_STREAM, changed from: These bitstreams MUST be decoded by the same MPEG-4 AVC/H.264 decoder as one stream. Changed to:

Errata Published*	Description
	These bitstreams MUST be encoded using the same MPEG-4 AVC/H.264 encoder and decoded by a single MPEG-4 AVC/H.264 decoder as one stream. Changed from: If no luma frame is present, then this field MUST be set to zero. Changed to: If no YUV420 frame is present, then this field MUST be set to zero. Changed from: 0x1 A YUV420 frame is contained in the avc420EncodedBitstream1 field, and no data is present in the avc420EncodedBitstream2 field. No Chroma420 frame is present. 0x2 A Chroma420 frame is contained in the avc420EncodedBitstream1 field, and no data is present in the avc420EncodedBitstream2 field. No YUV420 frame is present. Changed to: 0x1 A YUV420 frame is contained in the avc420EncodedBitstream1 field, and no data is present in the avc420EncodedBitstream2 field. No Chroma420 frame is present. The Chroma420 frame corresponding to the updates in the YUV420 frame is sent in a RFX_AVC444V2_BITMAP_STREAM message in subsequent frames if required. 0x2 A Chroma420 frame is contained in the avc420EncodedBitstream1 field, and no data is present in the avc420EncodedBitstream2 field. No YUV420 frame is present. The Chroma420 frame MUST be combined with the decoded AVC stream from previous frames.
2018/05/07	In Section 3.2.5.19, Sending an RDPGFX_CAPS_CONFIRM_PDU message, specific RDPGFX_CAPSET versions were removed from the description. Changed from: The structure and fields of the RDPGFX_CAPS_CONFIRM_PDU message are specified in section 2.2.2.19. The command fields MUST be populated in accordance with this description. The server MUST populate the capsSet field with a single instance of a correctly initialized RDPGFX_CAPSET_VERSION8 (section 2.2.3.1) or RDPGFX_CAPSET_VERSION81 (section 2.2.3.2) structure. Changed to: The structure and fields of the RDPGFX_CAPS_CONFIRM_PDU message are specified in section 2.2.2.19. The command fields MUST be populated in accordance with this description. The server MUST populate the capsSet field with a single instance of a correctly initialized capability set structure (section 2.2.3).
2018/04/23	In Section 1.5.2, Server Implementation Requirements, the list of capability sets that servers support that must be capable of processing the RDPGFX_QOE_FRAME_ACKNOWLEDGE_PDU message was updated. Changed from:

Errata Published*	Description
	Servers that support the RDPGFX_CAPSET_VERSION10 (section 2.2.3.3) capability set must be capable of processing the RDPGFX_QOE_FRAME_ACKNOWLEDGE_PDU (section 2.2.2.21) message. Changed to: Servers that support the RDPGFX_CAPSET_VERSION10 (section 2.2.3.3), RDPGFX_CAPSET_VERSION102 (section 2.2.3.5), RDPGFX_CAPSET_VERSION103 (section 2.2.3.6), RDPGFX_CAPSET_VERSION104 (section 2.2.3.7), or RDPGFX_CAPSET_VERSION105 (section 2.2.3.8) capability sets must be capable of processing the RDPGFX_QOE_FRAME_ACKNOWLEDGE_PDU (section 2.2.2.21) message.
2018/04/23	In Section 3.3.1.4, Bitmap Cache, the flag field list identifying the bitmap cache limit was updated to include RDPGFX_CAPSET_VERSION104 and RDPGFX_CAPSET_VERSION105. Changed from: The size of the bitmap cache is capped at 100 MB or 16 MB, depending on whether the RDPGFX_CAPS_FLAG_THINCLIENT (0x00000001) flag or RDPGFX_CAPS_FLAG_SMALL_CACHE (0x00000002) flag is specified in the flags field of an RDPGFX_CAPSET_VERSION8 (section 2.2.3.1), RDPGFX_CAPSET_VERSION81 (section 2.2.3.2), RDPGFX_CAPSET_VERSION10 (section 2.2.3.3), or RDPGFX_CAPSET_VERSION102 (section 2.2.3.5) structure, which is encapsulated in the server-to-client RDPGFX_CAPS_CONFIRM_PDU (section 2.2.2.19) message. Changed to: The size of the bitmap cache is capped at 100 MB or 16 MB, depending on whether the RDPGFX_CAPS_FLAG_THINCLIENT (0x00000001) flag or RDPGFX_CAPS_FLAG_SMALL_CACHE (0x00000002) flag is specified in the flags field of an RDPGFX_CAPSET_VERSION8 (section 2.2.3.1), RDPGFX_CAPSET_VERSION81 (section 2.2.3.2), RDPGFX_CAPSET_VERSION10 (section 2.2.3.3), RDPGFX_CAPSET_VERSION102 (section 2.2.3.5), RDPGFX_CAPSET_VERSION104 (section 2.2.3.7), or RDPGFX_CAPSET_VERSION105 (section 2.2.3.8) structure, which is encapsulated in the server-to-client RDPGFX_CAPS_CONFIRM_PDU (section 2.2.2.19) message.

*Date format: YYYY/MM/DD