

[MS-WMF]: Windows Metafile Format

This topic lists the Errata found in [MS-WMF] 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.

Errata below are for Protocol Document Version [V15.0 – 2018/09/12](#)

Errata Published*	Description																																																																																										
2021/02/08	<p>In section 2.2.2.13, PaletteEntry Object, revised the order of the fields.</p> <p>Changed from:</p> <p>The PaletteEntry Object defines the color and usage of an entry in a palette.</p> <table border="1" data-bbox="399 814 951 947"> <tr> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td> <td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td> <td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td> <td>3</td><td>1</td> </tr> <tr> <td colspan="4">Values</td> <td colspan="4">Blue</td> <td colspan="4">Green</td> <td colspan="4">Red</td> </tr> </table> <p>Values (1 byte): An 8-bit unsigned integer that defines how the palette entry is to be used. The Values field MUST be 0x00 or one of the values in the PaletteEntryFlag Enumeration (section 2.1.1.22) table.</p> <p>Blue (1 byte): An 8-bit unsigned integer that defines the blue intensity value for the palette entry.</p> <p>Green (1 byte): An 8-bit unsigned integer that defines the green intensity value for the palette entry.</p> <p>Red (1 byte): An 8-bit unsigned integer that defines the red intensity value for the palette entry.</p> <p>Changed to:</p> <p>The PaletteEntry Object defines the color and usage of an entry in a palette.</p> <table border="1" data-bbox="399 1392 951 1524"> <tr> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td> <td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td> <td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td> <td>3</td><td>1</td> </tr> <tr> <td colspan="4">Red</td> <td colspan="4">Green</td> <td colspan="4">Blue</td> <td colspan="4">Values</td> </tr> </table> <p>Red (1 byte): An 8-bit unsigned integer that defines the red intensity value for the palette entry.</p> <p>Green (1 byte): An 8-bit unsigned integer that defines the green intensity value for the palette entry.</p> <p>Blue (1 byte): An 8-bit unsigned integer that defines the blue intensity value for the palette entry.</p> <p>Values (1 byte): An 8-bit unsigned integer that defines how the palette entry is to be used. The Values field MUST be 0x00 or one of the values in the PaletteEntryFlag Enumeration (section 2.1.1.22) table.</p>	0	1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9	2	3	4	5	6	7	8	9	3	1	Values				Blue				Green				Red				0	1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9	2	3	4	5	6	7	8	9	3	1	Red				Green				Blue				Values			
0	1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9	2	3	4	5	6	7	8	9	3	1																																																															
Values				Blue				Green				Red																																																																															
0	1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9	2	3	4	5	6	7	8	9	3	1																																																															
Red				Green				Blue				Values																																																																															
2021/02/08	In Section 2.2.2.5, BitmapV5Header Object, revised the values used in the Intent field:																																																																																										

Errata Published*	Description				
	<p>Changed from:</p> <p>Intent (4 bytes): A 32-bit unsigned integer that defines the rendering intent for the DIB. This MUST be defined in the LogicalColorSpace Enumeration (section 2.1.1.14).</p> <p>Changed to:</p> <p>Intent (4 bytes): A 32-bit unsigned integer that defines the rendering intent for the DIB. This MUST be a value defined in the GamutMappingIntent Enumeration (section 2.1.1.11).</p>				
2021/02/08	<p>In Section 2.2.2.17, PolyPolygon Object, revised the definition of aPoints:</p> <p>Changed from:</p> <p>aPoints (variable): An array of 16-bit unsigned integers that define the coordinates of the polygons.</p> <p>Changed to:</p> <p>aPoints (variable): An array of PointS values that define the coordinates of the polygons. The length of the array is equal to the sum of all 16-bit integers in the aPointsPerPolygon array.</p> <p>In Section 2.3.3.15 META_POLYGON Record, revised casing and details of NumberOfPoints:</p> <p>Changed from:</p> <table border="1" data-bbox="415 1033 808 1083"> <tr> <td>RecordFunction</td> <td>NumberofPoints</td> </tr> </table> <p>Changed to:</p> <table border="1" data-bbox="415 1159 808 1209"> <tr> <td>RecordFunction</td> <td>NumberOfPoints</td> </tr> </table> <p>Changed from:</p> <p>NumberofPoints (2 bytes): A 16-bit signed integer that defines the number of points in the array.</p> <p>Changed to:</p> <p>NumberOfPoints (2 bytes): A 16-bit signed integer that defines the number of points in the array. This value must be greater than or equal to 2.</p>	RecordFunction	NumberofPoints	RecordFunction	NumberOfPoints
RecordFunction	NumberofPoints				
RecordFunction	NumberOfPoints				
2021/02/08	<p>In Section 2.3.3.5, META_EXTTEXTOUT RECORD, added string encoding details:</p> <p>Changed from:</p> <p>String (variable): A variable-length string that specifies the text to be drawn. The string does not need to be null-terminated, because StringLength specifies the length of the string. If the length is odd, an extra byte is placed after it so that the following member (optional Dx) is aligned on a 16-bit boundary.</p> <p>Changed to:</p>				

Errata Published*	Description																																				
	<p data-bbox="397 310 1404 514">String (variable): A variable-length string that specifies the text to be drawn. The string does not need to be null-terminated, because StringLength specifies the length of the string. If the length is odd, an extra byte is placed after it so that the following member (optional Dx) is aligned on a 16-bit boundary. The string will be decoded based on the font object currently selected into the playback device context. If a font matching the font object's specification is not found, the decoding is undefined. If a matching font is found that matches the charset specified in the font object, the string should be decoded with the codepages in the following table.</p> <table border="1" data-bbox="414 520 876 1430"> <thead> <tr> <th data-bbox="414 520 711 569">CharSet</th> <th data-bbox="711 520 876 569">CodePage ID</th> </tr> </thead> <tbody> <tr> <td data-bbox="414 569 711 617">ANSI_CHARSET</td> <td data-bbox="711 569 876 617">1252</td> </tr> <tr> <td data-bbox="414 617 711 665">OEM_CHARSET</td> <td data-bbox="711 617 876 665">437</td> </tr> <tr> <td data-bbox="414 665 711 714">SHIFTJIS_CHARSET</td> <td data-bbox="711 665 876 714">932</td> </tr> <tr> <td data-bbox="414 714 711 762">HANGEUL_CHARSET</td> <td data-bbox="711 714 876 762">949</td> </tr> <tr> <td data-bbox="414 762 711 810">JOHAB_CHARSET</td> <td data-bbox="711 762 876 810">1361</td> </tr> <tr> <td data-bbox="414 810 711 858">GB2312_CHARSET</td> <td data-bbox="711 810 876 858">936</td> </tr> <tr> <td data-bbox="414 858 711 907">CHINESEBIG5_CHARSET</td> <td data-bbox="711 858 876 907">950</td> </tr> <tr> <td data-bbox="414 907 711 955">HEBREW_CHARSET</td> <td data-bbox="711 907 876 955">1255</td> </tr> <tr> <td data-bbox="414 955 711 1003">ARABIC_CHARSET</td> <td data-bbox="711 955 876 1003">1256</td> </tr> <tr> <td data-bbox="414 1003 711 1052">GREEK_CHARSET</td> <td data-bbox="711 1003 876 1052">1253</td> </tr> <tr> <td data-bbox="414 1052 711 1100">TURKISH_CHARSET</td> <td data-bbox="711 1052 876 1100">1254</td> </tr> <tr> <td data-bbox="414 1100 711 1148">BALTIC_CHARSET</td> <td data-bbox="711 1100 876 1148">1257</td> </tr> <tr> <td data-bbox="414 1148 711 1197">EASTEUROPE_CHARSET</td> <td data-bbox="711 1148 876 1197">1250</td> </tr> <tr> <td data-bbox="414 1197 711 1245">RUSSIAN_CHARSET</td> <td data-bbox="711 1197 876 1245">1251</td> </tr> <tr> <td data-bbox="414 1245 711 1293">THAI_CHARSET</td> <td data-bbox="711 1245 876 1293">874</td> </tr> <tr> <td data-bbox="414 1293 711 1341">VIETNAMESE_CHARSET</td> <td data-bbox="711 1293 876 1341">1258</td> </tr> <tr> <td data-bbox="414 1341 711 1390">SYMBOL_CHARSET</td> <td data-bbox="711 1341 876 1390">42</td> </tr> </tbody> </table>	CharSet	CodePage ID	ANSI_CHARSET	1252	OEM_CHARSET	437	SHIFTJIS_CHARSET	932	HANGEUL_CHARSET	949	JOHAB_CHARSET	1361	GB2312_CHARSET	936	CHINESEBIG5_CHARSET	950	HEBREW_CHARSET	1255	ARABIC_CHARSET	1256	GREEK_CHARSET	1253	TURKISH_CHARSET	1254	BALTIC_CHARSET	1257	EASTEUROPE_CHARSET	1250	RUSSIAN_CHARSET	1251	THAI_CHARSET	874	VIETNAMESE_CHARSET	1258	SYMBOL_CHARSET	42
CharSet	CodePage ID																																				
ANSI_CHARSET	1252																																				
OEM_CHARSET	437																																				
SHIFTJIS_CHARSET	932																																				
HANGEUL_CHARSET	949																																				
JOHAB_CHARSET	1361																																				
GB2312_CHARSET	936																																				
CHINESEBIG5_CHARSET	950																																				
HEBREW_CHARSET	1255																																				
ARABIC_CHARSET	1256																																				
GREEK_CHARSET	1253																																				
TURKISH_CHARSET	1254																																				
BALTIC_CHARSET	1257																																				
EASTEUROPE_CHARSET	1250																																				
RUSSIAN_CHARSET	1251																																				
THAI_CHARSET	874																																				
VIETNAMESE_CHARSET	1258																																				
SYMBOL_CHARSET	42																																				
2021/02/08	<p data-bbox="397 1451 1193 1478">In Section 2.3, WMF Records, added information about record processing:</p> <p data-bbox="397 1520 560 1547">Changed from:</p> <p data-bbox="397 1589 1404 1717">When a WMFmetafile (2) is processed, the order in which graphics output is performed MUST be the same as the order of drawing records in the metafile (2). Thus, a given drawing command is always rendered on top of the renderings of preceding commands. The following packet definition specifies the generic structure of all WMF Records except Control Record Types (section 2.3.2).</p> <p data-bbox="397 1759 527 1787">Changed to:</p>																																				

Errata Published*	Description												
	<p>When a WMFmetafile (2) is processed, the order in which graphics output is performed MUST be the same as the order of drawing records in the metafile (2). Thus, a given drawing command is always rendered on top of the renderings of preceding commands.</p> <p>Implementations MUST ignore records with undocumented or unsupported record types and proceed to process the next valid record.</p> <p>The following packet definition specifies the generic structure of all WMF Records except Control Record Types (section 2.3.2).</p>												
2021/02/08	<p>In Section 2.3.2.3, META_PLACEABLE Record, clarified calculation of the value in the CheckSum field.</p> <p>Changed from:</p> <p>Checksum (2 bytes): A checksum for the previous 10 16-bit values in the header. This value can be used to determine whether the metafile has become corrupted.</p> <p>Changed to:</p> <p>Checksum (2 bytes): A checksum for the previous 10 16-bit values in the header. This value can be used to determine whether the metafile has become corrupted. The value is calculated by initializing the checksum to zero and then XORing it one at a time with the 10 16-bit values in the header.</p>												
2021/02/08	<p>In Section 2.2.1.2, Font Object, changed the length and description of the Facename field:</p> <p>Changed from:</p> <table border="1" data-bbox="415 1079 972 1178"> <tr> <td data-bbox="415 1079 521 1129">Quality</td> <td data-bbox="521 1079 712 1129">PitchAndFamily</td> <td data-bbox="712 1079 972 1129">Facename (Variable)</td> </tr> <tr> <td colspan="3" data-bbox="415 1129 972 1178">...</td> </tr> </table> <p>Changed to:</p> <table border="1" data-bbox="415 1289 979 1388"> <tr> <td data-bbox="415 1289 521 1339">Quality</td> <td data-bbox="521 1289 712 1339">PitchAndFamily</td> <td data-bbox="712 1289 979 1339">Facename (32 bytes)</td> </tr> <tr> <td colspan="3" data-bbox="415 1339 979 1388">...</td> </tr> </table> <p>Changed from:</p> <p>Facename (variable): A null-terminated string of 8-bit Latin-1 [ISO/IEC-8859-1] ANSI characters that specifies the typeface name of the font. The length of this string MUST NOT exceed 32 8-bit characters, including the terminating null.</p> <p>Changed to:</p> <p>Facename (32 bytes): A null-terminated string of up to 32 8-bit Latin-1 [ISO/IEC-8859-1] ANSI characters that specifies the typeface name of the font. Any characters following the terminating null are ignored.</p>	Quality	PitchAndFamily	Facename (Variable)	...			Quality	PitchAndFamily	Facename (32 bytes)	...		
Quality	PitchAndFamily	Facename (Variable)											
...													
Quality	PitchAndFamily	Facename (32 bytes)											
...													
2021/02/08	<p>In Section 2.3.3.5, META_EXTTEXTOUT Record, added information about when the Rectangle field is optional:</p>												

Errata Published*	Description
	<p>Changed from:</p> <p>Rectangle (8 bytes): An optional 8-byte Rect Object (section 2.2.2.18) that defines the dimensions, in logical coordinates, of a rectangle that is used for clipping, opaquing, or both.</p> <p>Changed to:</p> <p>Rectangle (8 bytes): An optional 8-byte Rect Object (section 2.2.2.18). When either ETO_CLIPPED, ETO_OPAQUE, or both are specified, the rectangle defines the dimensions, in logical coordinates, used for clipping, opaquing, or both. When neither ETO_CLIPPED nor ETO_OPAQUE is specified, the coordinates in Rectangle are ignored.</p>
2021/02/08	<p>In Section 2.3.2.3, META_PLACEABLE Record, added a cross-reference defining the BoundingBox:</p> <p>Changed from:</p> <p>BoundingBox (8 bytes): The rectangle in the playback context (or simply the destination rectangle), measured in logical units, for displaying the metafile. The size of a logical unit is specified by the Inch field.</p> <p>Changed to:</p> <p>BoundingBox (8 bytes): The rectangle in the playback context (or simply the destination rectangle), measured in logical units, for displaying the metafile. The size of a logical unit is specified by the Inch field. See section 2.2.2.18 for details about the structure of the BoundingBox field.</p>
2021/02/08	<p>The following sections have been changed.</p> <p>Section 2.1.1.17, MetafileEscapes Enumeration</p> <p>Section 2.1.1.23, PenStyle Enumeration</p> <p>Section 2.2.1.2, Font Object</p> <p>Section 2.3.3.15, META_POLYGON Record</p> <p>Section 2.3.6, Escape Record Types</p> <p>Section 2.3.6.4, CHECKJPEGFORMAT Record</p> <p>Section 2.3.6.5, CHECKPNGFORMAT Record</p> <p>Section 2.3.6.7, CLOSECHANNEL Record</p> <p>Section 2.3.6.8, DOWNLOADFACE Record</p> <p>Section 2.3.6.9, DOWNLOADHEADER Record</p> <p>Section 2.3.6.10, DRAWPATTERNRECT Record</p> <p>Section 2.3.6.14, EPSPRINTING Record</p> <p>Section 2.3.6.16, GETCOLORTABLE Record</p> <p>Section 2.3.6.17, GETDEVICEUNITS Record</p> <p>Section 2.3.6.18, GETEXTENDEDTEXTMETRICS Record</p> <p>Section 2.3.6.19, GETFACENAME Record</p> <p>Section 2.3.6.20, GETPAIRKERNTABLE Record</p> <p>Section 2.3.6.21, GETPHYSPPAGESIZE Record</p> <p>Section 2.3.6.22, GETPRINTINGOFFSET Record</p> <p>Section 2.3.6.24, GETSCALINGFACTOR Record</p> <p>Section 2.3.6.35, OPENCHANNEL Record</p>

Errata Published*	Description
	Section 2.3.6.36, QUERYDIBSUPPORT Record Section 2.3.6.37, QUERYESCSUPPORT Record Section 2.3.6.38, SETCOLORTABLE Record Section 2.3.6.39, SETCOPYCOUNT Record Section 2.3.6.40, SETLINECAP Record Section 2.3.6.41, SETLINEJOIN Record Section 2.3.6.42, SETMITERLIMIT Record Please see diff file

*Date format: YYYY/MM/DD