[MS-EMFPLUS]: Enhanced Metafile Format Plus Extensions

This topic lists the Errata found in the MS-EMFPLUS document 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 V16.0 – 2018/09/12.

Errata Published*	Description
2019/12/09	In Section 2.3.6.6, EmfPlusSetRenderingOrigin Record, changed from: x (4 bytes): An unsigned integer that defines the horizontal coordinate value of the rendering origin. y (4 bytes): An unsigned integer that defines the vertical coordinate value of the rendering origin. Changed to: x (4 bytes): A signed integer that defines the horizontal coordinate value of the rendering origin. y (4 bytes): A signed integer that defines the vertical coordinate value of the rendering origin.
2018/12/10	In this document several sections have been modified to reference [MS-LCID], the Windows Language Code Identifier (LCID) Reference. In Section 1.6, Versioning and Localization, changed from: Localization: EMF+ structures contain the following locale-specific data: • Language identifiers that correspond to natural languages in locales, including countries, geographical regions, and administrative districts. For details, see the LanguageIdentifier enumeration. Changed to: Localization: EMF+ structures contain the following locale-specific data: • Language identifiers that correspond to natural languages in locales, including countries, geographical regions, and administrative districts. For details, see [MS-LCID] section 2.1. In Section 2.1.1, Enumeration Constant Types, changed from: Name / Section / Description LineCapTypeLanguageIdentifier / 2.1.1.17 / Defines identifiers for natural languages in locales, including countries, geographical regions, and administrative

Errata Published*	Description
	<table><tr><td></td><td></td><td>districts. Defines types of line caps to use at the ends of lines that are drawn with graphics pens.</td></tr></table> Changed to: <table><tr><th>Name</th><th>Section</th><th>Description</th></tr><tr><td>LineCapType</td><td>2.1.1.17</td><td>Defines types of line caps to use at the ends of lines that are drawn with graphics pens.</td></tr></table> In Section 2.1.1.17, LanguageIdentifier Enumeration, the section title and introduction have been changed. Changed from: 2.1.1.17 LanguageIdentifier Enumeration The LanguageIdentifier enumeration defines identifiers for natural languages in locales, including countries, geographical regions, and administrative districts. Changed to: 2.1.1.17 LineCapType Enumeration The LineCapType enumeration defines types of line caps to use at the ends of lines that are drawn with graphics pens. In Section 2.2.2.23, EmfPlusLanguageIdentifier Object, changed from: … The encoded language identifier values are defined in the LanguageIdentifier enumeration. … Changed to: … The encoded LCID values are defined in [MS-LCID] section 2.2. … Section 2.1.3.2, Language Identifiers, has been removed.
2018/11/26	In Section 2.1.1, Enumeration Constant Types, the "WrapMode" enumeration has been added to the list of defined enumerations. Added: <table><tr><th>Name</th><th>Section</th><th>Description</th></tr><tr><td>…</td><td>…</td><td>…</td></tr></table>

<table>
<tr><th>Errata Published*</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td>WrapMode</td><td>2.1.1.34</td><td>Defines how the pattern from a texture or gradient brush is tiled across a shape or at shape boundaries.</td></tr>
</table>
In Section 2.1.2, Bit Flag Constant Types, the “PathPointType” enumeration has been added to the list of defined flags.
Added:
<table>
<tr><th>Name</th><th>Section</th><th>Description</th></tr>
<tr><td>…</td><td>…</td><td>…</td></tr>
<tr><td>PathPointType</td><td>2.1.2.6</td><td>Specifies the type properties of points on graphics paths.</td></tr>
<tr><td>…</td><td>…</td><td>…</td></tr>
</table>
In Section 2.3.8.1, EmfPlusSetTSClip, the name of the “Rects” field has been changed to “rects” throughout the section. For example, changed from:
rects (variable): An array of NumRects rectangles that define clipping areas. The format of this data is determined by the C bit in the Flags field.
The compression scheme for data in this record uses the following algorithm. Each point of each rectangle is encoded in either a single byte or 2 bytes. If the point is encoded in a single byte, the high bit (0x80) of the byte MUST be set, and the value is a signed number represented by the lower 7 bits. If the high bit is not set, then the value is encoded in 2 bytes, with the high-order byte encoded in the 7 lower bits of the first byte, and the low-order byte value encoded in the second byte.
Each point is encoded as the difference between the point in the current rect and the point in the previous rect. The bottom point of the rect is encoded as the difference between the bottom coordinate and the top coordinate on the current rect.
See section 2.3.8 for the specification of additional terminal server record types.
Changed to:
Rects (variable): An array of NumRects rectangles that define clipping areas. The format of this data is determined by the C bit in the Flags field.
The compression scheme for data in this record uses the following algorithm. Each point of each rectangle is encoded in either a single byte or 2 bytes. If the point is encoded in a single byte, the high bit (0x80) of the byte MUST be set, and the value is a signed number represented by the lower 7 bits. If the high bit is not set, then the value is encoded in 2 bytes, with the high-order byte encoded in the 7 lower bits of the first byte, and the low-order byte value encoded in the second byte.
Each point is encoded as the difference between the point in the current rectangle and the point in the previous rectangle. The bottom point of the rectangle is encoded as the difference between the bottom coordinate and the top coordinate on the current rectangle.
See section 2.3.8 for the specification of additional terminal server record types.
</td></tr>
</table>

*Date format: YYYY/MM/DD