**[MS-EMF]:**

**Enhanced Metafile Format**

Intellectual Property Rights Notice for Open Specifications Documentation

* **Technical Documentation.** Microsoft publishes Open Specifications documentation (“this documentation”) for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
* **Copyrights**. This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
* **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.
* **Patents**. Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft [Open Specifications Promise](https://go.microsoft.com/fwlink/?LinkId=214445) or the [Microsoft Community Promise](https://go.microsoft.com/fwlink/?LinkId=214448). If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting [iplg@microsoft.com](mailto:iplg@microsoft.com).
* **License Programs**. To see all of the protocols in scope under a specific license program and the associated patents, visit the [Patent Map](https://aka.ms/AA9ufj8).
* **Trademarks**. The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit [www.microsoft.com/trademarks](https://www.microsoft.com/trademarks).
* **Fictitious Names**. The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

**Reservation of Rights**. All other rights are reserved, and this notice does not grant any rights other than as specifically described above, whether by implication, estoppel, or otherwise.

**Tools**. The Open Specifications documentation does not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments, you are free to take advantage of them. Certain Open Specifications documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

**Support.** For questions and support, please contact [dochelp@microsoft.com](mailto:dochelp@microsoft.com).

**Revision Summary**

| Date | Revision History | Revision Class | Comments |
| --- | --- | --- | --- |
| 4/3/2007 | 0.01 | New | Version 0.01 release |
| 7/3/2007 | 1.0 | Major | MLonghorn+90 |
| 7/20/2007 | 2.0 | Major | Restructured record sections according to category; other updates. |
| 8/10/2007 | 2.1 | Minor | Clarified the meaning of the technical content. |
| 9/28/2007 | 2.2 | Minor | Clarified the meaning of the technical content. |
| 10/23/2007 | 3.0 | Major | Added new sections describing the EMR\_COMMENT\_EMFPLUS and EMR\_COMMENT\_EMFSPOOL records. |
| 11/30/2007 | 3.1 | Minor | Standardized art. |
| 1/25/2008 | 3.2 | Minor | Reconstructed record categories for clarity. |
| 3/14/2008 | 4.0 | Major | Abstract data model and Windows version-specific behavior added. |
| 5/16/2008 | 4.0.1 | Editorial | Changed language and formatting in the technical content. |
| 6/20/2008 | 4.1 | Minor | Clarified the meaning of the technical content. |
| 7/25/2008 | 4.1.1 | Editorial | Changed language and formatting in the technical content. |
| 8/29/2008 | 4.2 | Minor | Clarified the meaning of the technical content. |
| 10/24/2008 | 5.0 | Major | Updated and revised the technical content. |
| 12/5/2008 | 5.1 | Minor | Clarified the meaning of the technical content. |
| 1/16/2009 | 6.0 | Major | Updated and revised the technical content. |
| 2/27/2009 | 6.1 | Minor | Clarified the meaning of the technical content. |
| 4/10/2009 | 6.2 | Minor | Clarified the meaning of the technical content. |
| 5/22/2009 | 6.2.1 | Editorial | Changed language and formatting in the technical content. |
| 7/2/2009 | 6.3 | Minor | Clarified the meaning of the technical content. |
| 8/14/2009 | 6.4 | Minor | Clarified the meaning of the technical content. |
| 9/25/2009 | 6.5 | Minor | Clarified the meaning of the technical content. |
| 11/6/2009 | 6.5.1 | Editorial | Changed language and formatting in the technical content. |
| 12/18/2009 | 6.6 | Minor | Clarified the meaning of the technical content. |
| 1/29/2010 | 6.6.1 | Editorial | Changed language and formatting in the technical content. |
| 3/12/2010 | 6.7 | Minor | Clarified the meaning of the technical content. |
| 4/23/2010 | 6.7.1 | Editorial | Changed language and formatting in the technical content. |
| 6/4/2010 | 6.8 | Minor | Clarified the meaning of the technical content. |
| 7/16/2010 | 6.8 | None | No changes to the meaning, language, or formatting of the technical content. |
| 8/27/2010 | 6.8 | None | No changes to the meaning, language, or formatting of the technical content. |
| 10/8/2010 | 6.9 | Minor | Clarified the meaning of the technical content. |
| 11/19/2010 | 7.0 | Major | Updated and revised the technical content. |
| 1/7/2011 | 7.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 2/11/2011 | 7.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 3/25/2011 | 7.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 5/6/2011 | 7.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 6/17/2011 | 7.1 | Minor | Clarified the meaning of the technical content. |
| 9/23/2011 | 7.1 | None | No changes to the meaning, language, or formatting of the technical content. |
| 12/16/2011 | 8.0 | Major | Updated and revised the technical content. |
| 3/30/2012 | 8.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 7/12/2012 | 8.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 10/25/2012 | 8.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 1/31/2013 | 8.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 8/8/2013 | 9.0 | Major | Updated and revised the technical content. |
| 11/14/2013 | 9.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 2/13/2014 | 9.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 5/15/2014 | 10.0 | Major | Updated and revised the technical content. |
| 6/30/2015 | 11.0 | Major | Significantly changed the technical content. |
| 10/16/2015 | 11.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 7/14/2016 | 12.0 | Major | Significantly changed the technical content. |
| 6/1/2017 | 12.1 | Minor | Clarified the meaning of the technical content. |
| 9/15/2017 | 13.0 | Major | Significantly changed the technical content. |
| 12/1/2017 | 13.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 9/12/2018 | 14.0 | Major | Significantly changed the technical content. |
| 4/7/2021 | 15.0 | Major | Significantly changed the technical content. |
| 6/25/2021 | 16.0 | Major | Significantly changed the technical content. |
| 4/29/2022 | 17.0 | Major | Significantly changed the technical content. |

Table of Contents

[1 Introduction 10](#_Toc101868775)

[1.1 Glossary 10](#_Toc101868776)

[1.2 References 16](#_Toc101868777)

[1.2.1 Normative References 16](#_Toc101868778)

[1.2.2 Informative References 17](#_Toc101868779)

[1.3 Overview 17](#_Toc101868780)

[1.3.1 Metafile Structure 17](#_Toc101868781)

[1.3.2 Graphics Objects 19](#_Toc101868782)

[1.3.3 Byte Ordering 19](#_Toc101868783)

[1.4 Relationship to Protocols and Other Structures 19](#_Toc101868784)

[1.5 Applicability Statement 20](#_Toc101868785)

[1.6 Versioning and Localization 20](#_Toc101868786)

[1.7 Vendor-Extensible Fields 20](#_Toc101868787)

[2 Structures 21](#_Toc101868788)

[2.1 EMF Enumerations 21](#_Toc101868789)

[2.1.1 RecordType Enumeration 21](#_Toc101868790)

[2.1.2 ArcDirection Enumeration 28](#_Toc101868791)

[2.1.3 ArmStyle Enumeration 29](#_Toc101868792)

[2.1.4 BackgroundMode Enumeration 29](#_Toc101868793)

[2.1.5 ColorAdjustment Enumeration 30](#_Toc101868794)

[2.1.6 ColorMatchToTarget Enumeration 30](#_Toc101868795)

[2.1.7 ColorSpace Enumeration 30](#_Toc101868796)

[2.1.8 Contrast Enumeration 31](#_Toc101868797)

[2.1.9 DIBColors Enumeration 31](#_Toc101868798)

[2.1.10 EmrComment Enumeration 32](#_Toc101868799)

[2.1.11 ExtTextOutOptions Enumeration 32](#_Toc101868800)

[2.1.12 FamilyType Enumeration 33](#_Toc101868801)

[2.1.13 FloodFill Enumeration 34](#_Toc101868802)

[2.1.14 FormatSignature Enumeration 34](#_Toc101868803)

[2.1.15 GradientFill Enumeration 34](#_Toc101868804)

[2.1.16 GraphicsMode Enumeration 35](#_Toc101868805)

[2.1.17 HatchStyle Enumeration 36](#_Toc101868806)

[2.1.18 ICMMode Enumeration 36](#_Toc101868807)

[2.1.19 Illuminant Enumeration 36](#_Toc101868808)

[2.1.20 Letterform Enumeration 37](#_Toc101868809)

[2.1.21 MapMode Enumeration 38](#_Toc101868810)

[2.1.22 MetafileVersion Enumeration 39](#_Toc101868811)

[2.1.23 MidLine Enumeration 39](#_Toc101868812)

[2.1.24 ModifyWorldTransformMode Enumeration 40](#_Toc101868813)

[2.1.25 PenStyle Enumeration 40](#_Toc101868814)

[2.1.26 Point Enumeration 41](#_Toc101868815)

[2.1.27 PolygonFillMode Enumeration 42](#_Toc101868816)

[2.1.28 Proportion Enumeration 42](#_Toc101868817)

[2.1.29 RegionMode Enumeration 43](#_Toc101868818)

[2.1.30 SerifType Enumeration 43](#_Toc101868819)

[2.1.31 StockObject Enumeration 44](#_Toc101868820)

[2.1.32 StretchMode Enumeration 47](#_Toc101868821)

[2.1.33 StrokeVariation Enumeration 47](#_Toc101868822)

[2.1.34 Weight Enumeration 48](#_Toc101868823)

[2.1.35 XHeight Enumeration 48](#_Toc101868824)

[2.2 EMF Objects 49](#_Toc101868825)

[2.2.1 BitFIX28\_4 Object 49](#_Toc101868826)

[2.2.2 ColorAdjustment Object 49](#_Toc101868827)

[2.2.3 DesignVector Object 51](#_Toc101868828)

[2.2.4 EmrFormat Object 51](#_Toc101868829)

[2.2.5 EmrText Object 52](#_Toc101868830)

[2.2.6 EpsData Object 54](#_Toc101868831)

[2.2.7 GradientRectangle Object 55](#_Toc101868832)

[2.2.8 GradientTriangle Object 55](#_Toc101868833)

[2.2.9 Header Object 56](#_Toc101868834)

[2.2.10 HeaderExtension1 Object 57](#_Toc101868835)

[2.2.11 HeaderExtension2 Object 58](#_Toc101868836)

[2.2.12 LogBrushEx Object 58](#_Toc101868837)

[2.2.13 LogFont Object 59](#_Toc101868838)

[2.2.14 LogFontEx Object 61](#_Toc101868839)

[2.2.15 LogFontExDv Object 62](#_Toc101868840)

[2.2.16 LogFontPanose Object 62](#_Toc101868841)

[2.2.17 LogPalette Object 64](#_Toc101868842)

[2.2.18 LogPaletteEntry Object 64](#_Toc101868843)

[2.2.19 LogPen Object 65](#_Toc101868844)

[2.2.20 LogPenEx Object 65](#_Toc101868845)

[2.2.21 Panose Object 67](#_Toc101868846)

[2.2.22 PixelFormatDescriptor Object 68](#_Toc101868847)

[2.2.23 Point28\_4 Object 71](#_Toc101868848)

[2.2.24 RegionData Object 71](#_Toc101868849)

[2.2.25 RegionDataHeader Object 72](#_Toc101868850)

[2.2.26 TriVertex Object 72](#_Toc101868851)

[2.2.27 UniversalFontId Object 73](#_Toc101868852)

[2.2.28 XForm Object 74](#_Toc101868853)

[2.3 EMF Records 74](#_Toc101868854)

[2.3.1 Bitmap Record Types 75](#_Toc101868855)

[2.3.1.1 EMR\_ALPHABLEND Record 77](#_Toc101868856)

[2.3.1.2 EMR\_BITBLT Record 82](#_Toc101868857)

[2.3.1.3 EMR\_MASKBLT Record 84](#_Toc101868858)

[2.3.1.4 EMR\_PLGBLT Record 88](#_Toc101868859)

[2.3.1.5 EMR\_SETDIBITSTODEVICE Record 92](#_Toc101868860)

[2.3.1.6 EMR\_STRETCHBLT Record 94](#_Toc101868861)

[2.3.1.7 EMR\_STRETCHDIBITS Record 97](#_Toc101868862)

[2.3.1.8 EMR\_TRANSPARENTBLT Record 100](#_Toc101868863)

[2.3.2 Clipping Record Types 102](#_Toc101868864)

[2.3.2.1 EMR\_EXCLUDECLIPRECT Record 104](#_Toc101868865)

[2.3.2.2 EMR\_EXTSELECTCLIPRGN Record 105](#_Toc101868866)

[2.3.2.3 EMR\_INTERSECTCLIPRECT Record 105](#_Toc101868867)

[2.3.2.4 EMR\_OFFSETCLIPRGN Record 106](#_Toc101868868)

[2.3.2.5 EMR\_SELECTCLIPPATH Record 106](#_Toc101868869)

[2.3.3 Comment Record Types 107](#_Toc101868870)

[2.3.3.1 EMR\_COMMENT Record 108](#_Toc101868871)

[2.3.3.2 EMR\_COMMENT\_EMFPLUS Record 109](#_Toc101868872)

[2.3.3.3 EMR\_COMMENT\_EMFSPOOL Record 109](#_Toc101868873)

[2.3.3.4 EMR\_COMMENT\_PUBLIC Record Types 110](#_Toc101868874)

[2.3.3.4.1 EMR\_COMMENT\_BEGINGROUP Record 111](#_Toc101868875)

[2.3.3.4.2 EMR\_COMMENT\_ENDGROUP Record 112](#_Toc101868876)

[2.3.3.4.3 EMR\_COMMENT\_MULTIFORMATS Record 113](#_Toc101868877)

[2.3.3.4.4 EMR\_COMMENT\_WINDOWS\_METAFILE Record 114](#_Toc101868878)

[2.3.4 Control Record Types 115](#_Toc101868879)

[2.3.4.1 EMR\_EOF Record 116](#_Toc101868880)

[2.3.4.2 EMR\_HEADER Record Types 117](#_Toc101868881)

[2.3.4.2.1 EmfMetafileHeader Record 120](#_Toc101868882)

[2.3.4.2.2 EmfMetafileHeaderExtension1 Record 121](#_Toc101868883)

[2.3.4.2.3 EmfMetafileHeaderExtension2 Record 122](#_Toc101868884)

[2.3.5 Drawing Record Types 124](#_Toc101868885)

[2.3.5.1 EMR\_ANGLEARC Record 128](#_Toc101868886)

[2.3.5.2 EMR\_ARC Record 129](#_Toc101868887)

[2.3.5.3 EMR\_ARCTO Record 130](#_Toc101868888)

[2.3.5.4 EMR\_CHORD Record 131](#_Toc101868889)

[2.3.5.5 EMR\_ELLIPSE Record 132](#_Toc101868890)

[2.3.5.6 EMR\_EXTFLOODFILL Record 132](#_Toc101868891)

[2.3.5.7 EMR\_EXTTEXTOUTA Record 133](#_Toc101868892)

[2.3.5.8 EMR\_EXTTEXTOUTW Record 134](#_Toc101868893)

[2.3.5.9 EMR\_FILLPATH Record 135](#_Toc101868894)

[2.3.5.10 EMR\_FILLRGN Record 135](#_Toc101868895)

[2.3.5.11 EMR\_FRAMERGN Record 136](#_Toc101868896)

[2.3.5.12 EMR\_GRADIENTFILL Record 137](#_Toc101868897)

[2.3.5.13 EMR\_LINETO Record 139](#_Toc101868898)

[2.3.5.14 EMR\_PAINTRGN Record 140](#_Toc101868899)

[2.3.5.15 EMR\_PIE Record 140](#_Toc101868900)

[2.3.5.16 EMR\_POLYBEZIER Record 141](#_Toc101868901)

[2.3.5.17 EMR\_POLYBEZIER16 Record 142](#_Toc101868902)

[2.3.5.18 EMR\_POLYBEZIERTO Record 143](#_Toc101868903)

[2.3.5.19 EMR\_POLYBEZIERTO16 Record 144](#_Toc101868904)

[2.3.5.20 EMR\_POLYDRAW Record 145](#_Toc101868905)

[2.3.5.21 EMR\_POLYDRAW16 Record 146](#_Toc101868906)

[2.3.5.22 EMR\_POLYGON Record 147](#_Toc101868907)

[2.3.5.23 EMR\_POLYGON16 Record 148](#_Toc101868908)

[2.3.5.24 EMR\_POLYLINE Record 149](#_Toc101868909)

[2.3.5.25 EMR\_POLYLINE16 Record 150](#_Toc101868910)

[2.3.5.26 EMR\_POLYLINETO Record 150](#_Toc101868911)

[2.3.5.27 EMR\_POLYLINETO16 Record 151](#_Toc101868912)

[2.3.5.28 EMR\_POLYPOLYGON Record 152](#_Toc101868913)

[2.3.5.29 EMR\_POLYPOLYGON16 Record 153](#_Toc101868914)

[2.3.5.30 EMR\_POLYPOLYLINE Record 154](#_Toc101868915)

[2.3.5.31 EMR\_POLYPOLYLINE16 Record 156](#_Toc101868916)

[2.3.5.32 EMR\_POLYTEXTOUTA Record 156](#_Toc101868917)

[2.3.5.33 EMR\_POLYTEXTOUTW Record 158](#_Toc101868918)

[2.3.5.34 EMR\_RECTANGLE Record 159](#_Toc101868919)

[2.3.5.35 EMR\_ROUNDRECT Record 159](#_Toc101868920)

[2.3.5.36 EMR\_SETPIXELV Record 160](#_Toc101868921)

[2.3.5.37 EMR\_SMALLTEXTOUT Record 161](#_Toc101868922)

[2.3.5.38 EMR\_STROKEANDFILLPATH Record 162](#_Toc101868923)

[2.3.5.39 EMR\_STROKEPATH Record 163](#_Toc101868924)

[2.3.6 Escape Record Types 163](#_Toc101868925)

[2.3.6.1 EMR\_DRAWESCAPE Record 164](#_Toc101868926)

[2.3.6.2 EMR\_EXTESCAPE Record 165](#_Toc101868927)

[2.3.6.3 EMR\_NAMEDESCAPE Record 165](#_Toc101868928)

[2.3.7 Object Creation Record Types 166](#_Toc101868929)

[2.3.7.1 EMR\_CREATEBRUSHINDIRECT Record 168](#_Toc101868930)

[2.3.7.2 EMR\_CREATECOLORSPACE Record 169](#_Toc101868931)

[2.3.7.3 EMR\_CREATECOLORSPACEW Record 169](#_Toc101868932)

[2.3.7.4 EMR\_CREATEDIBPATTERNBRUSHPT Record 170](#_Toc101868933)

[2.3.7.5 EMR\_CREATEMONOBRUSH Record 172](#_Toc101868934)

[2.3.7.6 EMR\_CREATEPALETTE Record 173](#_Toc101868935)

[2.3.7.7 EMR\_CREATEPEN Record 174](#_Toc101868936)

[2.3.7.8 EMR\_EXTCREATEFONTINDIRECTW Record 175](#_Toc101868937)

[2.3.7.9 EMR\_EXTCREATEPEN Record 176](#_Toc101868938)

[2.3.8 Object Manipulation Record Types 177](#_Toc101868939)

[2.3.8.1 EMR\_COLORCORRECTPALETTE Record 179](#_Toc101868940)

[2.3.8.2 EMR\_DELETECOLORSPACE Record 180](#_Toc101868941)

[2.3.8.3 EMR\_DELETEOBJECT Record 180](#_Toc101868942)

[2.3.8.4 EMR\_RESIZEPALETTE Record 181](#_Toc101868943)

[2.3.8.5 EMR\_SELECTOBJECT Record 181](#_Toc101868944)

[2.3.8.6 EMR\_SELECTPALETTE Record 182](#_Toc101868945)

[2.3.8.7 EMR\_SETCOLORSPACE Record 182](#_Toc101868946)

[2.3.8.8 EMR\_SETPALETTEENTRIES Record 183](#_Toc101868947)

[2.3.9 OpenGL Record Types 184](#_Toc101868948)

[2.3.9.1 EMR\_GLSBOUNDEDRECORD Record 185](#_Toc101868949)

[2.3.9.2 EMR\_GLSRECORD Record 185](#_Toc101868950)

[2.3.10 Path Bracket Record Types 186](#_Toc101868951)

[2.3.11 State Record Types 187](#_Toc101868952)

[2.3.11.1 EMR\_COLORMATCHTOTARGETW Record 191](#_Toc101868953)

[2.3.11.2 EMR\_FORCEUFIMAPPING Record 192](#_Toc101868954)

[2.3.11.3 EMR\_INVERTRGN Record 192](#_Toc101868955)

[2.3.11.4 EMR\_MOVETOEX Record 193](#_Toc101868956)

[2.3.11.5 EMR\_PIXELFORMAT Record 194](#_Toc101868957)

[2.3.11.6 EMR\_RESTOREDC Record 194](#_Toc101868958)

[2.3.11.7 EMR\_SCALEVIEWPORTEXTEX Record 195](#_Toc101868959)

[2.3.11.8 EMR\_SCALEWINDOWEXTEX Record 195](#_Toc101868960)

[2.3.11.9 EMR\_SETARCDIRECTION Record 196](#_Toc101868961)

[2.3.11.10 EMR\_SETBKCOLOR Record 197](#_Toc101868962)

[2.3.11.11 EMR\_SETBKMODE Record 197](#_Toc101868963)

[2.3.11.12 EMR\_SETBRUSHORGEX Record 198](#_Toc101868964)

[2.3.11.13 EMR\_SETCOLORADJUSTMENT Record 198](#_Toc101868965)

[2.3.11.14 EMR\_SETICMMODE Record 199](#_Toc101868966)

[2.3.11.15 EMR\_SETICMPROFILEA Record 199](#_Toc101868967)

[2.3.11.16 EMR\_SETICMPROFILEW Record 200](#_Toc101868968)

[2.3.11.17 EMR\_SETLAYOUT Record 201](#_Toc101868969)

[2.3.11.18 EMR\_SETLINKEDUFIS Record 202](#_Toc101868970)

[2.3.11.19 EMR\_SETMAPMODE Record 202](#_Toc101868971)

[2.3.11.20 EMR\_SETMAPPERFLAGS Record 203](#_Toc101868972)

[2.3.11.21 EMR\_SETMITERLIMIT Record 203](#_Toc101868973)

[2.3.11.22 EMR\_SETPOLYFILLMODE Record 204](#_Toc101868974)

[2.3.11.23 EMR\_SETROP2 Record 204](#_Toc101868975)

[2.3.11.24 EMR\_SETSTRETCHBLTMODE Record 205](#_Toc101868976)

[2.3.11.25 EMR\_SETTEXTALIGN Record 205](#_Toc101868977)

[2.3.11.26 EMR\_SETTEXTCOLOR Record 206](#_Toc101868978)

[2.3.11.27 EMR\_SETTEXTJUSTIFICATION Record 206](#_Toc101868979)

[2.3.11.28 EMR\_SETVIEWPORTEXTEX Record 207](#_Toc101868980)

[2.3.11.29 EMR\_SETVIEWPORTORGEX Record 207](#_Toc101868981)

[2.3.11.30 EMR\_SETWINDOWEXTEX Record 208](#_Toc101868982)

[2.3.11.31 EMR\_SETWINDOWORGEX Record 208](#_Toc101868983)

[2.3.12 Transform Record Types 209](#_Toc101868984)

[2.3.12.1 EMR\_MODIFYWORLDTRANSFORM Record 210](#_Toc101868985)

[2.3.12.2 EMR\_SETWORLDTRANSFORM Record 211](#_Toc101868986)

[3 Structure Examples 212](#_Toc101868987)

[3.1 EMF Metafile Playback 212](#_Toc101868988)

[3.1.1 Abstract Data Model 212](#_Toc101868989)

[3.1.1.1 EMF Object Table 212](#_Toc101868990)

[3.1.1.2 Graphics Environment 214](#_Toc101868991)

[3.1.1.2.1 Regions 214](#_Toc101868992)

[3.1.1.2.2 Colors 215](#_Toc101868993)

[3.1.1.2.3 Text 215](#_Toc101868994)

[3.1.1.2.4 Drawing 216](#_Toc101868995)

[3.1.2 Byte Ordering 217](#_Toc101868996)

[3.2 EMF Metafile Example 218](#_Toc101868997)

[3.2.1 EMR\_HEADER Example 231](#_Toc101868998)

[3.2.2 EMR\_CREATEBRUSHINDIRECT Example 233](#_Toc101868999)

[3.2.3 EMR\_SELECTOBJECT Example 1 234](#_Toc101869000)

[3.2.4 EMR\_BITBLT Example 1 235](#_Toc101869001)

[3.2.5 EMR\_SELECTOBJECT Example 2 237](#_Toc101869002)

[3.2.6 EMR\_BITBLT Example 2 237](#_Toc101869003)

[3.2.7 EMR\_SETBKMODE Example 251](#_Toc101869004)

[3.2.8 EMR\_EXTCREATEFONTINDIRECTW Example 1 251](#_Toc101869005)

[3.2.9 EMR\_SELECTOBJECT Example 3 254](#_Toc101869006)

[3.2.10 EMR\_EXTTEXTOUTW Example 255](#_Toc101869007)

[3.2.11 EMR\_EXTCREATEFONTINDIRECTW Example 2 256](#_Toc101869008)

[3.2.12 EMR\_SELECTOBJECT Example 4 259](#_Toc101869009)

[3.2.13 EMR\_EXTCREATEFONTINDIRECTW Example 3 260](#_Toc101869010)

[3.2.14 EMR\_SELECTOBJECT Example 5 263](#_Toc101869011)

[3.2.15 EMR\_DELETEOBJECT Example 1 263](#_Toc101869012)

[3.2.16 EMR\_EXTCREATEFONTINDIRECTW Example 4 264](#_Toc101869013)

[3.2.17 EMR\_SELECTOBJECT Example 6 267](#_Toc101869014)

[3.2.18 EMR\_SELECTOBJECT Example 7 267](#_Toc101869015)

[3.2.19 EMR\_DELETEOBJECT Example 2 268](#_Toc101869016)

[3.2.20 EMR\_DELETEOBJECT Example 3 268](#_Toc101869017)

[3.2.21 EMR\_SELECTOBJECT Example 8 268](#_Toc101869018)

[3.2.22 EMR\_EOF Example 269](#_Toc101869019)

[4 Security Considerations 270](#_Toc101869020)

[5 Appendix A: Product Behavior 271](#_Toc101869021)

[6 Change Tracking 278](#_Toc101869022)

[7 Index 279](#_Toc101869023)

# Introduction

[**Enhanced metafile format (EMF)**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) is a file format that is used to store portable representations of graphical images. EMF [**metafiles**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7) contain sequential records that are parsed and processed to render the stored image on any output device.

Sections 1.7 and 2 of this specification are normative. All other sections and examples in this specification are informative.

## Glossary

This document uses the following terms:

**28.4 bit FIX notation**: A notation for representing the location of a point on a device surface to within one-sixteenth of a pixel. Each point coordinate is a 32-bit value, of which the 28 higher-order bits are the signed integral part and the 4 lower-order bits are the unsigned fractional part, in one-sixteenth units of distance. For example, the number 0x0000003C is a coordinate value of 3.75, because the fractional part is 12 sixteenths, or 0.75.

**alpha transparency**: An alpha value is a transparency value represented by a number between zero and one. Each pixel has an alpha value that represents its level of transparency, which is multiplied by the color values to get the final value. Each pixel has an alpha value that represents its level of transparency.

**American National Standards Institute (ANSI) character set**: A character set defined by a code page approved by the American National Standards Institute (ANSI). The term "ANSI" as used to signify Windows code pages is a historical reference and a misnomer that persists in the Windows community. The source of this misnomer stems from the fact that the Windows code page 1252 was originally based on an ANSI draft, which became International Organization for Standardization (ISO) Standard 8859-1 [[ISO/IEC-8859-1]](https://go.microsoft.com/fwlink/?LinkId=90689). In Windows, the ANSI character set can be any of the following code pages: 1252, 1250, 1251, 1253, 1254, 1255, 1256, 1257, 1258, 874, 932, 936, 949, or 950. For example, "ANSI application" is usually a reference to a non-[**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) or code-page-based application. Therefore, "ANSI character set" is often misused to refer to one of the character sets defined by a Windows code page that can be used as an active system code page; for example, character sets defined by code page 1252 or character sets defined by code page 950. Windows is now based on [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8), so the use of ANSI character sets is strongly discouraged unless they are used to interoperate with legacy applications or legacy data.

**anisotropic**: Refers to the properties of an image, such as the scaling of logical units to device units, which are not the same regardless of the direction (x-axis versus y-axis) that is measured. Contrast with [**isotropic**](#gt_716e3252-cd28-451e-be10-bde8793f0534).

**ASCII**: The American Standard Code for Information Interchange (ASCII) is an 8-bit character-encoding scheme based on the English alphabet. ASCII codes represent text in computers, communications equipment, and other devices that work with text. ASCII refers to a single 8-bit ASCII character or an array of 8-bit ASCII characters with the high bit of each character set to zero.

**aspect ratio**: The ratio that is computed by dividing the width of a pixel on a given output device by its height.

**baseline**: The imaginary line to which the bottom of the lowercase "x" character in a font [**typeface**](#gt_2fff9b0b-4bc1-4d05-81ef-8dde59fd9789) is aligned.

**Bezier curve**: A type of curve, defined by a mathematical formula and a number of points greater than or equal to two, which is used in computer graphics and in the mathematical field of numeric analysis. A cubic [**Bezier curve**](#gt_b12730d5-b22e-4027-ba59-ff53749022a5) is defined by four points: two endpoints and two control points. The curve does not pass through the control points, but the control points act like magnets, pulling the curve in certain directions and influencing the way the curve bends. With multiple [**Bezier curves**](#gt_b12730d5-b22e-4027-ba59-ff53749022a5), the endpoint of one is the starting point of the next.

**big-endian**: Multiple-byte values that are byte-ordered with the most significant byte stored in the memory location with the lowest address.

**bitmap**: A collection of structures that contain a representation of a graphical image, a [**logical palette**](#gt_43e451cc-1df4-4abc-bb24-aac32f5ab6f3), dimensions and other information.

**Boolean**: An operation or expression that can be evaluated only as either true or false.

**brightness**: The relative lightness or darkness of an image, or of a particular color in an image.

**cell height**: A vertical measure of font size, which is the sum of the font height and internal leading. It might not be the same as the distance between two lines of text.

**Color correction**: Altering the colors in an image in order to print or display it such that the colors correctly match reality.

**color gamut**: The entire range of colors that is available on a particular graphics output device such as a display or printer.

**color mapping**: The process of associating integer color indices with color channel values.

**color matching**: The conversion of a color, sent from its original color space, to its visually closest color in the destination color space. See also [**Image Color Management (ICM)**](#gt_fcd5ee55-57ff-45d6-b51e-2ccba47329e3).

**color plane**: In [**bitmap**](#gt_26a17e58-eeba-4210-81b9-8bbadf7ee952) graphics, all pixel information for a single color. See also color channel.

**color profile**: A file that contains information about how to convert colors in the color space and the color gamut of a specific device into a device-independent color space. A device-specific color profile is called a "device profile". For more information on using color and device profiles, see [[MSDN-UDP]](https://go.microsoft.com/fwlink/?LinkId=90151).

**color proofing**: The process of previewing, or proofing colors, which were developed on one device, on a different device.

**color space**: A mapping of color components to a multidimensional coordinate system. The number of dimensions is generally two, three, or four. For example, if colors are expressed as a combination of the three components red, green, and blue, a three-dimensional space is sufficient to describe all possible colors. If transparency is considered one of the components of an RGB color, four dimensions are appropriate.

**color table**: An array of data that maps pixel values into a [**color space**](#gt_5d3fb5ea-c686-4d3b-b0ff-aef33fe1aee5).

**colorfulness**: A concept referring to the perceived [**intensity**](#gt_f44dd201-e056-4467-bf8a-2dc1f4586a44) of a specific color, the difference between a color against gray.

**compositing**: The process that takes place during image rendering, which combines color data from multiple graphics region.

**contrast**: The relative difference between lightness and darkness in an area of an image.

**coordinate space**: A space based on Cartesian coordinates, which provides a means of specifying the location of each point in the space. A two-dimensional coordinate space requires two axes that are perpendicular and equal in length. Three two-dimensional coordinate spaces are generally used to describe an output surface: world, page, and device. To scale device-independent output for a particular physical device, a rectangular area in the world or page coordinate space is mapped into the device coordinate space using a [**transform**](#gt_fdca4a96-8e04-4dc0-acad-1d01e1384a5b)

**design vector**: A set of specific values for the font axes of a [**multiple master**](#gt_fef5ee7a-ad0c-4395-a1ac-f130405eec03) font.

**device context**: A collection of properties and objects that defines a dynamic environment for processes on a device. For graphics output, properties include brush style, line style, text layout, foreground and background colors, and [**mapping mode**](#gt_6e998e8d-4394-4263-8632-503730f0abae); and objects include a brush, pen, font, [**palette**](#gt_dca5cd91-8d01-424b-9722-c9e818562ee5), [**region**](#gt_6f13fe5f-8747-4ae0-9375-814bf0528197), and transform matrix. Multiple device contexts can exist simultaneously, but a single device context specifies the environment for graphics output at a particular point in time.

**device space**: The output space for graphics [**transforms**](#gt_fdca4a96-8e04-4dc0-acad-1d01e1384a5b). It usually refers to the client area of an application window; however, it can also include the entire desktop, a complete window, or a page of printer or plotter paper. Physical device space dimensions vary according to the dimensions set by the display, printer, or plotter technology.

**device-independent bitmap (DIB)**: A container for bitmapped graphics, which specifies characteristics of the [**bitmap**](#gt_26a17e58-eeba-4210-81b9-8bbadf7ee952) such that it can be created using one application and loaded and displayed in another application, while retaining an identical appearance.

**dithering**: A form of digital halftoning.

**ducking**: A ducking font is one that has been designed to be short enough to fit under diacritical marks or accent marks.

**em size**: A measure of font size, which is the [**cell height**](#gt_b97118e3-2175-4af5-b1f9-462e57501090) minus the [**internal leading**](#gt_9efb809f-982e-4f1e-9dc8-89a65f22c732). An "em" is a term that has been used historically as a unit of typeset size.

**encapsulated PostScript (EPS)**: A file of [**PostScript**](#gt_31971c63-f974-47de-8de9-6b76d093890d) raw data that describes the appearance of a single page. Although EPS data can describe text, graphics, and images; the primary purpose of an EPS file is to be encapsulated within another [**PostScript**](#gt_31971c63-f974-47de-8de9-6b76d093890d) page definition.

**enhanced metafile format (EMF)**: A file format that supports the device-independent definitions of images.

**enhanced metafile format plus extensions (EMF+)**: A file format that supports the device-independent definitions of images.

**enhanced metafile spool format (EMFSPOOL)**: A format that specifies a structure of [**enhanced metafile format (EMF)**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) records used for defining application and device-independent printer spool files.

**font axis**: A property of font design that can assume a linear range of values. In general, a font has multiple axes. For example, a font may define an axis for [**weight**](#gt_fc6de273-4e8b-4f97-89d0-3dc1b4be9e8d), along which range the possible values for that property.

**font hinting**: The use of mathematical operations to manipulate the appearance of an outline font so that it lines up with a [**rasterized**](#gt_e4616703-011e-4bcf-842a-fbd0f778d76c) grid. At small resolutions, with or without anti-aliasing, hinting is critical for producing clear, legible text for human readers.

**font mapper**: An operating system component that maps specified font attributes to available, installed fonts on the system.

**gamma correction**: In digital imaging, the process of changing the brightness, contrast, or color balance of an image by assigning new values (different colors) to gray or color tones.

**Graphics Device Interface (GDI)**: An API, supported on 16-bit and 32-bit versions of the operating system, that supports graphics operations and image manipulation on logical graphics objects.

**Graphics Device Interface, Extended (GDI+)**: A Windows API, supported on 32-bit and 64-bit versions of the operating system, that extends [**GDI**](#gt_6c550e84-9a9c-46ef-8716-7260fe29cae3) to include support for Bezier curves, gradient brushes, image effects, and [**EMF+**](#gt_41a5b58b-b050-4001-85ec-d6ed56379ee2) metafiles.

**halftone**: A color representation consisting of a discrete gray or tone level.

**Image Color Management (ICM)**: Technology that ensures that a color image, graphic, or text object is rendered as closely as possible to its original intent on any device despite differences in imaging technologies and color capabilities between devices.

**inclusive-inclusive**: When referring to the bounds of a rectangle that consist of two coordinates—one coordinate for one corner and the other coordinate for the opposite corner [**inclusive-inclusive**](#gt_c706d1bc-8878-4743-9b94-e9643c5269e4) means that the coordinates are part of the rectangle. If not [**inclusive-inclusive**](#gt_c706d1bc-8878-4743-9b94-e9643c5269e4), the coordinates are not part of the rectangle and instead are one logical unit outside the bounds of the rectangle along both coordinate axes.

**intensity**: The magnitude of a component color in the color space.

**internal leading**: The amount of space inside a character cell, within the bounds set by the font ascent. Accent marks can occur in this area.

**isotropic**: Refers to the properties of an image, such as the scaling of logical units to device units, which are the same regardless of the direction (x-axis versus y-axis) that is measured. Contrast with [**anisotropic**](#gt_8356f4cd-ad99-4fb4-a10e-6aeeb1712c26).

**Joint Photographic Experts Group (JPEG)**: A raster graphics file format for displaying high-resolution color graphics. JPEG graphics apply a user-specified compression scheme that can significantly reduce the file sizes of photo-realistic color graphics. A higher level of compression results in lower quality, whereas a lower level of compression results in higher quality. JPEG-format files have a .jpg or .jpeg file name extension.

**line cap**: The shape that is used at the end of a line drawn by a graphics pen.

**line join**: The shape to use at the intersection of two lines drawn by a graphics pen.

**little-endian**: Multiple-byte values that are byte-ordered with the least significant byte stored in the memory location with the lowest address.

**logical palette**: A [**palette**](#gt_dca5cd91-8d01-424b-9722-c9e818562ee5) that defines colors as device-independent values. Unlike the [**system palette**](#gt_874d5ba8-6fd5-4c63-97f3-f35cfeda95fe), which has predefined, device-specific color definitions, a logical palette contains color values that can be defined entirely by an application. A logical palette entry is mapped to the [**system palette**](#gt_874d5ba8-6fd5-4c63-97f3-f35cfeda95fe) entry in order for the custom colors to appear when the application is run.

**mapping mode**: The way in which logical (device-independent) coordinates are mapped to [**device space**](#gt_25d586c7-f440-45c6-b965-5c025bea9fc0) (device-specific) coordinates. It also specifies the orientation of the axes and size of the units used for drawing operations.

**metafile**: A sequence of record structures that store an image in an application-independent format. Metafile records contain drawing commands, object definitions, and configuration settings. When a metafile is processed, the stored image can be rendered on a display, output to a printer or plotter, stored in memory, or saved to a file or stream.

**miter length**: At the intersection of two lines, the distance from the intersection of the line walls on the inside of the [**line join**](#gt_4fd3ebb8-54f8-47e8-aded-399a08ad873e) to the intersection of the line walls on the outside of the [**line join**](#gt_4fd3ebb8-54f8-47e8-aded-399a08ad873e). The miter length can be large when the angle between two lines is small. If the miter length of the join of an intersection exceeds a specified limit, the join can be beveled to keep it within the limit of the join of the intersection.

**monoscopic**: The property of an image that conveys a lack of the illusion of depth, as if the image were two-dimensional.

**multiple master**: A font technology that is a variation of the [**PostScript**](#gt_31971c63-f974-47de-8de9-6b76d093890d) [**Type 1 font**](#gt_8539bd4b-4321-40c3-b4d2-aefe3c846e4e) format. Because multiple master fonts are outline fonts, changing their size does not affect the quality of their output. Multiple master technology supports the creation of an unlimited number of custom variations of a font, called instances, as well as the emulation of [**typefaces**](#gt_2fff9b0b-4bc1-4d05-81ef-8dde59fd9789) that might not be present on the user's system.

**OpenGL**: A software API for graphics hardware that supports the rendering of multidimensional graphical objects. The Microsoft implementation of OpenGL for the Windows operating system provides industry-standard graphics software for creating high-quality still and animated three-dimensional color images. See [[OPENGL]](https://go.microsoft.com/fwlink/?LinkId=90241) for further information.

**OpenType**: A [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8)-based font technology that is an extension to [**TrueType**](#gt_855615c9-5378-459f-8d58-20f640afed6e) and [**Type 1 font**](#gt_8539bd4b-4321-40c3-b4d2-aefe3c846e4e) technologies. OpenType allows [**PostScript**](#gt_31971c63-f974-47de-8de9-6b76d093890d) and [**TrueType**](#gt_855615c9-5378-459f-8d58-20f640afed6e) glyph definitions to reside in a common container format.

**original equipment manufacturer (OEM) character set**: A character encoding used where the mappings between characters is dependent upon the code page configured on the machine, typically by the manufacturer.

**packed DIB**: A [**device-independent bitmap (DIB)**](#gt_3a6934d5-cc59-4ab9-a54e-e062a09a4091) in which the bit array immediately follows the header (see [[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.9).

**page space**: A logical coordinate system used for graphics operations. It is determined by the [**mapping mode**](#gt_6e998e8d-4394-4263-8632-503730f0abae). Page space is defined with device-independent units, such as pixels.

**palette**: An array of values, each element of which contains the definition of a color. The color elements in a palette are often indexed so that clients can refer to the colors, each of which can occupy 24 bits or more, by a number that requires less storage space.

**PANOSE**: A classification system for font [**typefaces**](#gt_2fff9b0b-4bc1-4d05-81ef-8dde59fd9789) that is based on certain specific visual characteristics of the font, including [**weight**](#gt_fc6de273-4e8b-4f97-89d0-3dc1b4be9e8d) (emphasis) and serif style.

**path**: A graphics object that is a container for a series of line and curve segments, and [**regions**](#gt_6f13fe5f-8747-4ae0-9375-814bf0528197) in an image.

**path bracket**: A series of [**paths**](#gt_2cd71385-2d9c-4ab8-bf4a-7b258816d613) that composes a larger figure. A path bracket specifies the current [**path**](#gt_2cd71385-2d9c-4ab8-bf4a-7b258816d613) that is defined in the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45).

**pitch**: A property of a font that describes the horizontal density of characters in a font; that is, the number of characters that can fit in a given unit of space. When all the characters in a font have the same width, the font is called "fixed-pitch"; if characters can have various widths, the font is "variable-pitch". Times New Roman is a variable-pitch font; it is easy to see that the characters in the font have different widths. For example, the width of a lowercase "i" is visibly less than the width of an uppercase "W".

**playback device context**: The [**device context**](#gt_92e112c6-791f-48dd-8345-e65a35eef388) that defines the current graphics state during playback of the [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7). Although the data in a [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7) can be device-independent, playback is always associated with an output device with specific properties, such as resolution, color support, and so on.

**Portable Network Graphics (PNG)**: A bitmap graphics file format that uses lossless data compression and supports variable transparency of images (alpha channels) and control of image brightness on different computers (gamma correction). PNG-format files have a .png file name extension.

**PostScript**: A page description language developed by Adobe Systems that is primarily used for printing documents on laser printers. It is the standard for desktop publishing.

**print server**: A machine that hosts the print system and all its different components.

**printer driver**: The interface component between the operating system and the printer device. It is responsible for processing the application data into a page description language (PDL) that can be interpreted by the printer device.

**raster operation**: The process of combining the bits in a source [**bitmap**](#gt_26a17e58-eeba-4210-81b9-8bbadf7ee952) with the bits in a destination [**bitmap**](#gt_26a17e58-eeba-4210-81b9-8bbadf7ee952) and in a specified pattern, to achieve a particular graphical output.

**rasterized font**: A font produced with rasterization. Such fonts are not scalable; they define glyph [**bitmaps**](#gt_26a17e58-eeba-4210-81b9-8bbadf7ee952) at specific sizes. Because of this, the appearance of rasterized fonts does not improve in proportion to the resolution of an output device. When magnified, the visual quality of a rasterized font decreases significantly compared to a [**vector font**](#gt_4795d9b6-f3e9-4435-874e-bb6882ea1f0f).

**rasterizer**: A program that converts geometric shapes into matrixes of discrete pixel settings on a graphics object such as a font.

**red-green-blue (RGB)**: A color model that describes color information in terms of the red (R), green (G), and blue (B) intensities in a color.

**red-green-blue-alpha (RGBA)**: A color model that describes color information in terms of the red (R), green (G), blue (B), and alpha (A) intensities that comprise a color.

**reflection transform**: A [**transform**](#gt_fdca4a96-8e04-4dc0-acad-1d01e1384a5b) that is used to create a mirror image of an object with respect to either the horizontal or vertical axis.

**region**: A graphics object that is nonrectilinear in shape and is defined by an array of scanlines.

**rotation**: A transform that is used to rotate an object. When rotation occurs, the points that make up the object are rotated with respect to the coordinate space origin.

**rotation transform**: A [**transform**](#gt_fdca4a96-8e04-4dc0-acad-1d01e1384a5b) that is used to rotate an object. When rotation occurs, the points that make up an object are rotated with respect to the [**coordinate space**](#gt_84184077-d0d9-41e0-acaf-6596aa066813) origin.

**scaling transform**: A [**transform**](#gt_fdca4a96-8e04-4dc0-acad-1d01e1384a5b) that is used to stretch or compress an object horizontally or vertically.

**shear transform**: A [**transform**](#gt_fdca4a96-8e04-4dc0-acad-1d01e1384a5b) that is used to shear or cut an object. There are two components of a shear [**transform**](#gt_fdca4a96-8e04-4dc0-acad-1d01e1384a5b): The first alters the vertical lines in an object, and the second alters the horizontal lines.

**stereoscopic**: The property of an image that gives the illusion of depth, as if the image were three-dimensional. The pixels that compose such an image can include a [**color plane**](#gt_78d62be3-d07a-447e-ab51-fa8be5560fe0) that is designed to add that illusion.

**stock object**: A predefined graphics object. Stock objects are standard, commonly used objects, such as a black brush and pen. The set of predefined stock objects is specified in [MS-EMF] section 2.1.31. Stock objects are neither created nor deleted.

**system palette**: The [**palette**](#gt_dca5cd91-8d01-424b-9722-c9e818562ee5) that is actually in use to reproduce colors on a device such as a computer screen. A system palette has predefined, device-specific colors that are used by default, so that individual applications do not have to set them up.

**tint**: The amount of a neutral color, such as black or white, that is mixed with another color. Changing the tint increases or decreases the lightness and saturation, and leaves the hue unchanged.

**transform**: An algorithm that transforms the size, orientation, and shape of objects that are copied from one [**coordinate space**](#gt_84184077-d0d9-41e0-acaf-6596aa066813) into another. Although a transform affects an object as a whole, it is applied to each point, or to each line, in the object.

**translation transform**: A [**transform**](#gt_fdca4a96-8e04-4dc0-acad-1d01e1384a5b) that is used to shift each point in an object vertically, horizontally, or both, by a specified amount.

**TrueType**: A scalable font technology that renders fonts for both the printer and the screen. Each TrueType font contains its own algorithms for converting printer outlines into screen [**bitmaps**](#gt_26a17e58-eeba-4210-81b9-8bbadf7ee952), which means both the outline and [**bitmap**](#gt_26a17e58-eeba-4210-81b9-8bbadf7ee952) information is [**rasterized**](#gt_e4616703-011e-4bcf-842a-fbd0f778d76c) from the same font data. The lower-level language embedded within the TrueType font allows great flexibility in its design. Both TrueType and [**Type 1 font**](#gt_8539bd4b-4321-40c3-b4d2-aefe3c846e4e) technologies are part of the [**OpenType**](#gt_5e824598-05ae-44e2-99bd-78c347279ad9) format.

**Type 1 font**: A public, standard type format originally developed for use with [**PostScript**](#gt_31971c63-f974-47de-8de9-6b76d093890d) printers. Type 1 fonts contain two components—the outline font, used for printing; and the [**bitmap**](#gt_26a17e58-eeba-4210-81b9-8bbadf7ee952) font set, used for screen display.

**typeface**: The primary design of a set of printed characters such as Courier, Helvetica, and Times Roman. The terms typeface and font are sometimes used interchangeably. A font is the particular implementation and variation of the typeface such as normal, bold, or italics. The distinguishing characteristic of a typeface is often the presence or absence of serifs.

**Unicode**: A character encoding standard developed by the Unicode Consortium that represents almost all of the written languages of the world. The [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) standard [[UNICODE5.0.0/2007]](https://go.microsoft.com/fwlink/?LinkId=154659) provides three forms (UTF-8, UTF-16, and UTF-32) and seven schemes (UTF-8, UTF-16, UTF-16 BE, UTF-16 LE, UTF-32, UTF-32 LE, and UTF-32 BE).

**UTF-16LE (Unicode Transformation Format, 16-bits, little-endian)**: The encoding scheme specified in [UNICODE5.0.0/2007] section 2.6 for encoding Unicode characters as a sequence of 16-bit codes, each encoded as two 8-bit bytes with the least-significant byte first.

**vector font**: A font that is defined with geometrical primitives such as points, lines, curves, and polygons, which are all based on mathematical equations instead of collections of discrete pixel settings. Vector fonts can be rendered in high quality at arbitrary sizes. Outline fonts are vector fonts. Contrast with [**rasterized fonts**](#gt_e4616703-011e-4bcf-842a-fbd0f778d76c).

**weight**: The property of a font that specifies the degree of emphasis or boldness of the characters.

**Windows Color System (WCS)**: Color management technology that ensures a color image, graphic, or text object is rendered as closely as possible to its original intent on any device, despite differences in imaging technologies and color capabilities between devices. WCS is a superset of [**ICM**](#gt_fcd5ee55-57ff-45d6-b51e-2ccba47329e3) APIs and functionality and includes a variety of new functions that provide significant improvements in flexibility, transparency, predictability, and extensibility for vendors. Windows NT 3.1 operating system Windows NT 3.5 operating system, Windows NT 3.51 operating system, Windows 98 operating system, and Windows Millennium Edition operating system do not support [**WCS**](#gt_e4f9c4a4-c195-43d0-b189-f9727ec8a2b3) color management.

**Windows metafile format (WMF)**: A file format used by Windows that supports the definition of images, including a format for clip art in word-processing documents.

**world space**: The most abstract logical [**coordinate space**](#gt_84184077-d0d9-41e0-acaf-6596aa066813) for graphics [**transforms**](#gt_fdca4a96-8e04-4dc0-acad-1d01e1384a5b). It allows [**scaling**](#gt_f3f5f638-16f6-4159-a3f1-9d845b1196f5), [**translation**](#gt_4d42abda-3b31-4828-bc4c-684f2de4e2af), [**rotation**](#gt_5cdd5e5b-fe3d-46e6-9019-50483463a76a), [**shearing**](#gt_de5c1f5c-2ce7-4e79-9be9-a01b09175f5b), and [**reflection**](#gt_bbc1ce5c-0fd6-4fde-abb3-40525a95147d).

**MAY, SHOULD, MUST, SHOULD NOT, MUST NOT:** These terms (in all caps) are used as defined in [[RFC2119]](https://go.microsoft.com/fwlink/?LinkId=90317). All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

## References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the [Errata](https://go.microsoft.com/fwlink/?linkid=850906).

### Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact [dochelp@microsoft.com](mailto:dochelp@microsoft.com). We will assist you in finding the relevant information.

[ISO/IEC-8859-1] International Organization for Standardization, "Information Technology -- 8-Bit Single-Byte Coded Graphic Character Sets -- Part 1: Latin Alphabet No. 1", ISO/IEC 8859-1, 1998, [http://www.iso.org/iso/home/store/catalogue\_tc/catalogue\_detail.htm?csnumber=28245](https://go.microsoft.com/fwlink/?LinkId=90689)

**Note** There is a charge to download the specification.

[MS-DTYP] Microsoft Corporation, "[Windows Data Types](%5bMS-DTYP%5d.pdf#Section_cca2742956894a16b2b49325d93e4ba2)".

[MS-WMF] Microsoft Corporation, "[Windows Metafile Format](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2)".

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, [https://www.rfc-editor.org/rfc/rfc2119.html](https://go.microsoft.com/fwlink/?LinkId=90317)

[UNICODE] The Unicode Consortium, "The Unicode Consortium Home Page", [http://www.unicode.org/](https://go.microsoft.com/fwlink/?LinkId=90550)

### Informative References

[DEVMODE] Microsoft Corporation, "DEVMODE structure", [http://msdn.microsoft.com/en-us/library/dd183565(VS.85).aspx](https://go.microsoft.com/fwlink/?LinkId=89844)

[MS-EMFPLUS] Microsoft Corporation, "[Enhanced Metafile Format Plus Extensions](%5bMS-EMFPLUS%5d.pdf#Section_5f92c78964f246b59ed415a9bb0946c6)".

[MS-EMFSPOOL] Microsoft Corporation, "[Enhanced Metafile Spool Format](%5bMS-EMFSPOOL%5d.pdf#Section_3d8cd6cc528742e8925f4a53afd04534)".

[MSDN-GDI+] Microsoft Corporation, "GDI+", [https://docs.microsoft.com/en-us/windows/desktop/gdiplus/-gdiplus-gdi-start](https://go.microsoft.com/fwlink/?LinkId=90013)

[MSDN-WindowsGDI] Microsoft Corporation, "Windows GDI", [http://msdn.microsoft.com/en-us/library/dd145203.aspxx](https://go.microsoft.com/fwlink/?LinkId=94735)

[MSDN-WRLDPGSPC] Microsoft Corporation, "World-Space to Page-Space Transformations", [http://msdn.microsoft.com/en-us/library/ms532657.aspx](https://go.microsoft.com/fwlink/?LinkId=90169)

[OPENGL] Segal, M. and Akeley, K., "The OpenGL Graphics System: A Specification, Version 2.1", December 2006, [http://www.opengl.org/registry/doc/glspec21.20061201.pdf](https://go.microsoft.com/fwlink/?LinkId=90241)

## Overview

An [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7) is a sequence of variable-length records that contain drawing commands, object definitions, and graphics properties to render a graphical image on any output device.

When an EMF metafile is "played back" on a specific device, the resulting image maintains the same dimensions, shape, proportions, color, and so on, as the original, regardless of how the metafile was created. At any given point in the playback, the state of the graphics environment is defined by the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45).

### Metafile Structure

An EMF metafile begins with a EMR\_HEADER record (section [2.3.4.2](#Section_de081cd7351f4cc2830bd03fb55e89ab)), which includes the metafile version, its size, the resolution of the device on which the picture was created, and it ends with an EMR\_EOF record (section [2.3.4.1](#Section_3f47fde00e6b40c187f3f4129af03aa1)). Between them are records that specify the rendering of the image.

The following diagram depicts the structures of different versions of EMF metafiles, which are:

* **Original**: The original version of EMF defined a file container for the device-independent specification of images.[<1>](#Appendix_A_1" \o "Product behavior note 1)
* **Extension 1**: The second version and first extension added a pixel format record and support for [**OpenGL**](#gt_216becb9-a800-4a44-8206-2ec2231c9c67) commands, enhancing the device independence and flexibility of EMF.
* **Extension 2**: The third version and second extension added the capability to measure distances on device surfaces by using the Metric system, enhancing the accuracy and scalability of EMF.

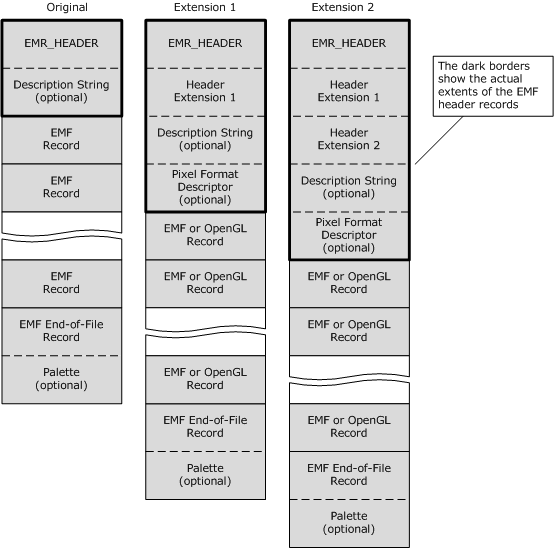


Figure 1: EMF metafile versions

EMF metafiles can be considered to have three parts:

* **EMF header**: An EMR\_HEADER record (section 2.3.4.2), possibly with extensions. It contains information concerning the structure and contents of the metafile, including an optional description string and pixel format descriptor.
* **EMF records:** A sequence of drawing orders, property settings, and object definitions (section [2.3](#Section_e0137630f3ad492cbde9e68866e255ba)). At least one record is present, not counting the header and end-of-file.
* **EMF end-of-file**: The EMR\_EOF record (section 2.3.4.1), which is the last record in the EMF metafile.

EMF records are contiguous because the information that is available for traversing the metafile from record to record depends on it. From any given EMF record, except EMR\_EOF, the length of that record can be used to move to the next record in the metafile.

### Graphics Objects

Graphics objects are used by the drawing and painting operations specified in an [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7). They are created by object creation records (section [2.3.7](#Section_7d914464d5464393bddb6d1e747fca18)) during the metafile playback and saved for later use.

When needed, a graphics object is be retrieved and selected into the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45) by using an EMR\_SELECTOBJECT record (section [2.3.8.5](#Section_145b063d5f9641feb7ae1e615b2bc2bf)).

The types of reusable objects include:

* Brushes ([2.2.12](#Section_6d790f8d51654746bac33443915c2071))
* Color spaces ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) sections 2.2.2.11 and 2.2.2.12)
* Fonts (sections [2.2.13](#Section_1cd7cb7a1a7c48d2b4a4c218f9d12100), [2.2.14](#Section_71516231003445f2af78c463b98db4af), [2.2.15](#Section_595fcaaebc3741489afe859e81ca7f76), and [2.2.16](#Section_c68ffff81c324398ac812cb6f0cd7a7c))
* Palettes (section [2.2.17](#Section_758f047d765e424c9204a833b7b4e527))
* Pens (sections [2.2.19](#Section_93ce3f4537ac4affb6e82f6db054c4c4) and [2.2.20](#Section_5b67b3eeea004f809b732959804381be))

[**Stock objects**](#gt_b1328cb9-acd8-47f8-9896-fe985af2d6d6) are used until graphics objects are created.

### Byte Ordering

Data in [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7) records is stored in [**little-endian**](#gt_079478cb-f4c5-4ce5-b72b-2144da5d2ce7) format.

Some computer architectures number bytes in a binary word from left to right, which is referred to as [**big-endian**](#gt_6f6f9e8e-5966-4727-8527-7e02fb864e7e). The byte numbering used for bitfields in this specification is big-endian. Other architectures number the bytes in a binary word from right to left, which is referred to as little-endian. The byte numbering used for enumerations, objects, and records in this specification is little-endian.

Using the big-endian and little-endian methods, the number 0x12345678 would be stored as shown in the following table.

| Byte order | Byte 0 | Byte 1 | Byte 2 | Byte 3 |
| --- | --- | --- | --- | --- |
| Big-endian | 0x12 | 0x34 | 0x56 | 0x78 |
| Little-endian | 0x78 | 0x56 | 0x34 | 0x12 |

For an example of how the use of the big-endian and little-endian methods can affect the compatibility of applications, see section [3.1.2](#Section_26465dd0ab564321b5c96d93bc179945).

## Relationship to Protocols and Other Structures

Several related [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7) formats can be used together for device-independent printing. Their relationships are:

* [**Enhanced metafile spool format (EMFSPOOL)**](#gt_3bd355a2-4faa-40ab-bc79-0d158eba43b6) records ([[MS-EMFSPOOL]](%5bMS-EMFSPOOL%5d.pdf#Section_3d8cd6cc528742e8925f4a53afd04534) section 2.2) can contain EMF records.
* [**Enhanced metafile format (EMF)**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) records (section [2.3](#Section_e0137630f3ad492cbde9e68866e255ba)) can contain EMF+ records.[<2>](#Appendix_A_2" \o "Product behavior note 2)
* [**Enhanced metafile format plus extensions (EMF+)**](#gt_41a5b58b-b050-4001-85ec-d6ed56379ee2) records ([[MS-EMFPLUS]](%5bMS-EMFPLUS%5d.pdf#Section_5f92c78964f246b59ed415a9bb0946c6) section 2.3) can contain [**Windows metafile format (WMF)**](#gt_48849cf6-d55c-47e5-b041-13b65854de2b) records ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.3).

This is illustrated qualitatively in the following figure.

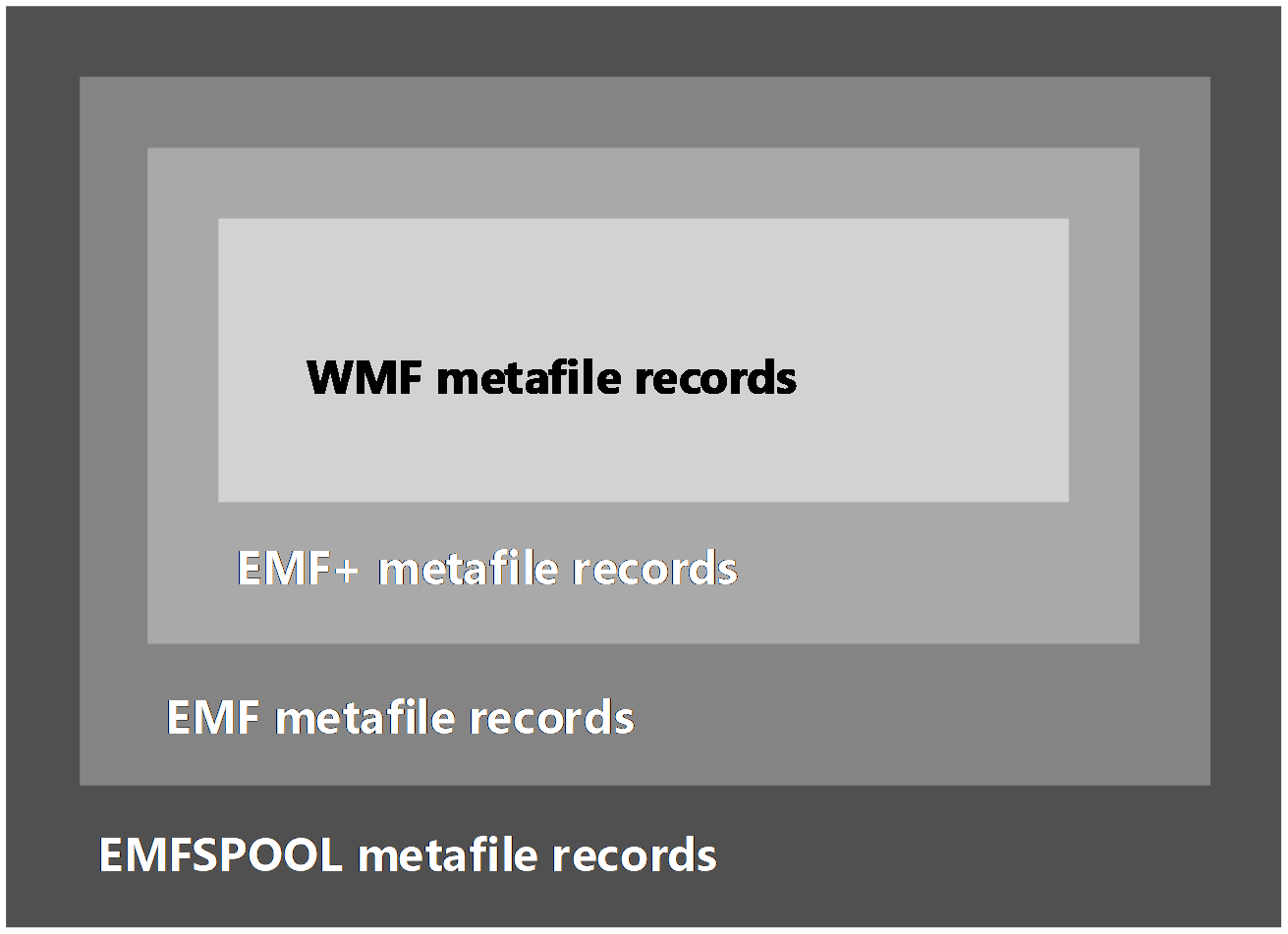


Figure 2: Relationships of metafile record types

## Applicability Statement

[**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) [**metafiles**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7) are portable containers for graphical output. The graphics format supported by EMF is applicable to rendering output on all devices, including displays, printers, and plotters.

## Versioning and Localization

This specification covers versioning issues in the following areas:

**Versioning:** The [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) structure has been revised twice. The different versions are:

* Original: The first version of the EMF structure, supporting records that define drawing commands and graphics objects.
* Extension 1: Added support for [**OpenGL**](#gt_216becb9-a800-4a44-8206-2ec2231c9c67) records and an optional internal pixel format descriptor.
* Extension 2: Added the capability of measuring display dimensions in micrometers.

**Localization:** This structure defines no locale-specific processes or data.

## Vendor-Extensible Fields

[**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) [**metafiles**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7) define a mechanism for the encapsulation of arbitrary vendor-defined data. The EMR\_COMMENT record (section [2.3.3.1](#Section_e081b202429d4c34b21ca0ad501858a6)) can contain arbitrary private data that is unknown to EMF. This data is meaningful only to applications that can decode the format of the data.

# Structures

The following sections specify [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) enumerations, objects, and records.

All character strings specified in this section are encoded in [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) [**UTF16-LE**](#gt_f25550c9-f84f-4eb2-8156-14794a7e3059) format [[UNICODE]](https://go.microsoft.com/fwlink/?LinkId=90550) unless stated otherwise.

Unreferenced intrinsic types are defined in [[MS-DTYP]](%5bMS-DTYP%5d.pdf#Section_cca2742956894a16b2b49325d93e4ba2).

## EMF Enumerations

### RecordType Enumeration

The **RecordType** enumeration defines values that uniquely identify records in an [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7). These values are specified in the **Type** fields of EMF records (section [2.3](#Section_e0137630f3ad492cbde9e68866e255ba)).

1. typedef enum
2. {
3. EMR\_HEADER = 0x00000001,
4. EMR\_POLYBEZIER = 0x00000002,
5. EMR\_POLYGON = 0x00000003,
6. EMR\_POLYLINE = 0x00000004,
7. EMR\_POLYBEZIERTO = 0x00000005,
8. EMR\_POLYLINETO = 0x00000006,
9. EMR\_POLYPOLYLINE = 0x00000007,
10. EMR\_POLYPOLYGON = 0x00000008,
11. EMR\_SETWINDOWEXTEX = 0x00000009,
12. EMR\_SETWINDOWORGEX = 0x0000000A,
13. EMR\_SETVIEWPORTEXTEX = 0x0000000B,
14. EMR\_SETVIEWPORTORGEX = 0x0000000C,
15. EMR\_SETBRUSHORGEX = 0x0000000D,
16. EMR\_EOF = 0x0000000E,
17. EMR\_SETPIXELV = 0x0000000F,
18. EMR\_SETMAPPERFLAGS = 0x00000010,
19. EMR\_SETMAPMODE = 0x00000011,
20. EMR\_SETBKMODE = 0x00000012,
21. EMR\_SETPOLYFILLMODE = 0x00000013,
22. EMR\_SETROP2 = 0x00000014,
23. EMR\_SETSTRETCHBLTMODE = 0x00000015,
24. EMR\_SETTEXTALIGN = 0x00000016,
25. EMR\_SETCOLORADJUSTMENT = 0x00000017,
26. EMR\_SETTEXTCOLOR = 0x00000018,
27. EMR\_SETBKCOLOR = 0x00000019,
28. EMR\_OFFSETCLIPRGN = 0x0000001A,
29. EMR\_MOVETOEX = 0x0000001B,
30. EMR\_SETMETARGN = 0x0000001C,
31. EMR\_EXCLUDECLIPRECT = 0x0000001D,
32. EMR\_INTERSECTCLIPRECT = 0x0000001E,
33. EMR\_SCALEVIEWPORTEXTEX = 0x0000001F,
34. EMR\_SCALEWINDOWEXTEX = 0x00000020,
35. EMR\_SAVEDC = 0x00000021,
36. EMR\_RESTOREDC = 0x00000022,
37. EMR\_SETWORLDTRANSFORM = 0x00000023,
38. EMR\_MODIFYWORLDTRANSFORM = 0x00000024,
39. EMR\_SELECTOBJECT = 0x00000025,
40. EMR\_CREATEPEN = 0x00000026,
41. EMR\_CREATEBRUSHINDIRECT = 0x00000027,
42. EMR\_DELETEOBJECT = 0x00000028,
43. EMR\_ANGLEARC = 0x00000029,
44. EMR\_ELLIPSE = 0x0000002A,
45. EMR\_RECTANGLE = 0x0000002B,
46. EMR\_ROUNDRECT = 0x0000002C,
47. EMR\_ARC = 0x0000002D,
48. EMR\_CHORD = 0x0000002E,
49. EMR\_PIE = 0x0000002F,
50. EMR\_SELECTPALETTE = 0x00000030,
51. EMR\_CREATEPALETTE = 0x00000031,
52. EMR\_SETPALETTEENTRIES = 0x00000032,
53. EMR\_RESIZEPALETTE = 0x00000033,
54. EMR\_REALIZEPALETTE = 0x00000034,
55. EMR\_EXTFLOODFILL = 0x00000035,
56. EMR\_LINETO = 0x00000036,
57. EMR\_ARCTO = 0x00000037,
58. EMR\_POLYDRAW = 0x00000038,
59. EMR\_SETARCDIRECTION = 0x00000039,
60. EMR\_SETMITERLIMIT = 0x0000003A,
61. EMR\_BEGINPATH = 0x0000003B,
62. EMR\_ENDPATH = 0x0000003C,
63. EMR\_CLOSEFIGURE = 0x0000003D,
64. EMR\_FILLPATH = 0x0000003E,
65. EMR\_STROKEANDFILLPATH = 0x0000003F,
66. EMR\_STROKEPATH = 0x00000040,
67. EMR\_FLATTENPATH = 0x00000041,
68. EMR\_WIDENPATH = 0x00000042,
69. EMR\_SELECTCLIPPATH = 0x00000043,
70. EMR\_ABORTPATH = 0x00000044,
71. EMR\_COMMENT = 0x00000046,
72. EMR\_FILLRGN = 0x00000047,
73. EMR\_FRAMERGN = 0x00000048,
74. EMR\_INVERTRGN = 0x00000049,
75. EMR\_PAINTRGN = 0x0000004A,
76. EMR\_EXTSELECTCLIPRGN = 0x0000004B,
77. EMR\_BITBLT = 0x0000004C,
78. EMR\_STRETCHBLT = 0x0000004D,
79. EMR\_MASKBLT = 0x0000004E,
80. EMR\_PLGBLT = 0x0000004F,
81. EMR\_SETDIBITSTODEVICE = 0x00000050,
82. EMR\_STRETCHDIBITS = 0x00000051,
83. EMR\_EXTCREATEFONTINDIRECTW = 0x00000052,
84. EMR\_EXTTEXTOUTA = 0x00000053,
85. EMR\_EXTTEXTOUTW = 0x00000054,
86. EMR\_POLYBEZIER16 = 0x00000055,
87. EMR\_POLYGON16 = 0x00000056,
88. EMR\_POLYLINE16 = 0x00000057,
89. EMR\_POLYBEZIERTO16 = 0x00000058,
90. EMR\_POLYLINETO16 = 0x00000059,
91. EMR\_POLYPOLYLINE16 = 0x0000005A,
92. EMR\_POLYPOLYGON16 = 0x0000005B,
93. EMR\_POLYDRAW16 = 0x0000005C,
94. EMR\_CREATEMONOBRUSH = 0x0000005D,
95. EMR\_CREATEDIBPATTERNBRUSHPT = 0x0000005E,
96. EMR\_EXTCREATEPEN = 0x0000005F,
97. EMR\_POLYTEXTOUTA = 0x00000060,
98. EMR\_POLYTEXTOUTW = 0x00000061,
99. EMR\_SETICMMODE = 0x00000062,
100. EMR\_CREATECOLORSPACE = 0x00000063,
101. EMR\_SETCOLORSPACE = 0x00000064,
102. EMR\_DELETECOLORSPACE = 0x00000065,
103. EMR\_GLSRECORD = 0x00000066,
104. EMR\_GLSBOUNDEDRECORD = 0x00000067,
105. EMR\_PIXELFORMAT = 0x00000068,
106. EMR\_DRAWESCAPE = 0x00000069,
107. EMR\_EXTESCAPE = 0x0000006A,
108. EMR\_SMALLTEXTOUT = 0x0000006C,
109. EMR\_FORCEUFIMAPPING = 0x0000006D,
110. EMR\_NAMEDESCAPE = 0x0000006E,
111. EMR\_COLORCORRECTPALETTE = 0x0000006F,
112. EMR\_SETICMPROFILEA = 0x00000070,
113. EMR\_SETICMPROFILEW = 0x00000071,
114. EMR\_ALPHABLEND = 0x00000072,
115. EMR\_SETLAYOUT = 0x00000073,
116. EMR\_TRANSPARENTBLT = 0x00000074,
117. EMR\_GRADIENTFILL = 0x00000076,
118. EMR\_SETLINKEDUFIS = 0x00000077,
119. EMR\_SETTEXTJUSTIFICATION = 0x00000078,
120. EMR\_COLORMATCHTOTARGETW = 0x00000079,
121. EMR\_CREATECOLORSPACEW = 0x0000007A
122. } RecordType;

**EMR\_HEADER:** This record defines the start of the metafile and specifies its characteristics; its contents, including the dimensions of the embedded image; the number of records in the metafile; and the resolution of the device on which the embedded image was created. These values make it possible for the metafile to be device-independent.

**EMR\_POLYBEZIER:** This record defines one or more [**Bezier curves**](#gt_b12730d5-b22e-4027-ba59-ff53749022a5). Cubic Bezier curves are defined using specified endpoints and control points, and are stroked with the current pen.

**EMR\_POLYGON:** This record defines a polygon consisting of two or more vertexes connected by straight lines. The polygon is outlined by using the current pen and filled by using the current brush and polygon fill mode. The polygon is closed automatically by drawing a line from the last vertex to the first.

**EMR\_POLYLINE:** This record defines a series of line segments by connecting the points in the specified array.

**EMR\_POLYBEZIERTO:** This record defines one or more Bezier curves based upon the current drawing position.

**EMR\_POLYLINETO:** This record defines one or more straight lines based upon the current drawing position. A line is drawn from the current drawing position to the first point specified by the points field by using the current pen. For each additional line, drawing is performed from the ending point of the previous line to the next point specified by points.

**EMR\_POLYPOLYLINE:** This record defines multiple series of connected line segments. The line segments are drawn by using the current pen. The figures formed by the segments are not filled. The current position is neither used nor updated by this record.

**EMR\_POLYPOLYGON:** This record defines a series of closed polygons. Each polygon is outlined by using the current pen and filled by using the current brush and polygon fill mode. The polygons defined by this record can overlap.

**EMR\_SETWINDOWEXTEX:** This record defines the window extent.

**EMR\_SETWINDOWORGEX:** This record defines the window origin.

**EMR\_SETVIEWPORTEXTEX:** This record defines the viewport extent.

**EMR\_SETVIEWPORTORGEX:** This record defines the viewport origin.

**EMR\_SETBRUSHORGEX:** This record defines the origin of the current brush.

**EMR\_EOF:** This record indicates the end of the metafile.

**EMR\_SETPIXELV:** This record defines the color of the pixel at the specified logical coordinates.

**EMR\_SETMAPPERFLAGS:** This record specifies parameters for the process of matching logical fonts to physical fonts, which is performed by the [**font mapper**](#gt_3c6f410b-e1f3-4841-8254-6f45433425f1).

**EMR\_SETMAPMODE:** This record defines the [**mapping mode**](#gt_6e998e8d-4394-4263-8632-503730f0abae), which defines the unit of measure used to transform [**page space**](#gt_f530f123-858a-4af7-b626-adea0aeaa75a) units into [**device space**](#gt_25d586c7-f440-45c6-b965-5c025bea9fc0) units, and defines the orientation of the device's X and Y axes.

**EMR\_SETBKMODE:** This record defines the background mix mode, which is used with text, hatched brushes, and pen styles that are not solid lines.

**EMR\_SETPOLYFILLMODE:** This record defines polygon fill mode.

**EMR\_SETROP2:** This record defines binary [**raster operation**](#gt_772681cd-b9a5-440c-8021-b6219c70f6fc) mode.

**EMR\_SETSTRETCHBLTMODE:** This record defines [**bitmap**](#gt_26a17e58-eeba-4210-81b9-8bbadf7ee952) stretch mode.

**EMR\_SETTEXTALIGN:** This record defines text alignment.

**EMR\_SETCOLORADJUSTMENT:** This record defines the color adjustment values using the specified values.

**EMR\_SETTEXTCOLOR:** This record defines the current text color.

**EMR\_SETBKCOLOR:** This record defines the background color.

**EMR\_OFFSETCLIPRGN:** This record redefines the current clipping [**region**](#gt_6f13fe5f-8747-4ae0-9375-814bf0528197) by the specified offsets.

**EMR\_MOVETOEX:** This record defines coordinates of the new drawing position in logical units.

**EMR\_SETMETARGN:** This record intersects the current clipping region with the current metaregion and saves the combined region as the new current metaregion.

**EMR\_EXCLUDECLIPRECT:** This record defines a new clipping region that consists of the current clipping region intersected with the specified rectangle.

**EMR\_INTERSECTCLIPRECT:** This record defines a new clipping region from the intersection of the current clipping region and the specified rectangle.

**EMR\_SCALEVIEWPORTEXTEX:** This record redefines the viewport using the ratios formed by the specified multiplicands and divisors.

**EMR\_SCALEWINDOWEXTEX:** This record redefines the window using the ratios formed by the specified multiplicands and divisors.

**EMR\_SAVEDC:** This record saves the current state of the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45) (section [3.1](#Section_50ae16a09d1748018bf1e28e9533a889)) in an array of states saved by preceding **EMR\_SAVEDC** records if any.

**EMR\_RESTOREDC:** This record restores the playback device context to the specified state, which was saved by a preceding **EMR\_SAVEDC** record (section [2.3.11](#Section_8f8a4df2f8d74d39afc094e19f524652)).

**EMR\_SETWORLDTRANSFORM:** This record defines a two-dimensional linear [**transform**](#gt_fdca4a96-8e04-4dc0-acad-1d01e1384a5b) between [**world space**](#gt_baadb707-1343-413b-9b36-2496b06e9bd7) and page space [[MSDN-WRLDPGSPC]](https://go.microsoft.com/fwlink/?LinkId=90169).

**EMR\_MODIFYWORLDTRANSFORM:** This record redefines the world transform by using the specified mode.

**EMR\_SELECTOBJECT:** This record selects an object in the playback device context, which is identified by its index in the EMF object table (section [3.1.1.1](#Section_e4fa4e6390964cdcb77685e2a1e4e1f4)).

**EMR\_CREATEPEN:** This record defines a logical pen (section [2.2.19](#Section_93ce3f4537ac4affb6e82f6db054c4c4)) that has the specified style, width, and color.

**EMR\_CREATEBRUSHINDIRECT:** This record defines a logical brush for filling figures in graphics operations.

**EMR\_DELETEOBJECT:** This record deletes a graphics object, clearing its index in the EMF object table.

**EMR\_ANGLEARC:** This record defines a line segment of an arc. The line segment is drawn from the current drawing position to the beginning of the arc. The arc is drawn along the perimeter of a circle with the given radius and center. The length of the arc is defined by the given start and sweep angles.

**EMR\_ELLIPSE:** This record defines an ellipse. The center of the ellipse is the center of the specified bounding rectangle. The ellipse is outlined by using the current pen and is filled by using the current brush.

**EMR\_RECTANGLE:** This record defines a rectangle. The rectangle is outlined by using the current pen and filled by using the current brush.

**EMR\_ROUNDRECT:** This record defines a rectangle with rounded corners. The rectangle is outlined by using the current pen and filled by using the current brush.

**EMR\_ARC:** This record defines an elliptical arc.

**EMR\_CHORD:** This record defines a chord, which is a region bounded by the intersection of an ellipse and a line segment, called a secant. The chord is outlined by using the current pen and filled by using the current brush.

**EMR\_PIE:** This record defines a pie-shaped wedge bounded by the intersection of an ellipse and two radials. The pie is outlined by using the current pen and filled by using the current brush.

**EMR\_SELECTPALETTE:** This record selects a LogPalette object (section [2.2.17](#Section_758f047d765e424c9204a833b7b4e527)) into the playback device context, identifying it by its index in the EMF object table.

**EMR\_CREATEPALETTE:** This record defines a LogPalette object.

**EMR\_SETPALETTEENTRIES:** This record defines [**RGB**](#gt_2c716d3a-e60b-4e52-bbb0-2fdeb298003b) color values in a range of entries in a LogPalette object.

**EMR\_RESIZEPALETTE:** This record increases or decreases the size of a [**logical palette**](#gt_43e451cc-1df4-4abc-bb24-aac32f5ab6f3).

**EMR\_REALIZEPALETTE:** This record maps entries from the current logical palette to the [**system palette**](#gt_874d5ba8-6fd5-4c63-97f3-f35cfeda95fe).

**EMR\_EXTFLOODFILL:** This record fills an area of the display surface with the current brush.

**EMR\_LINETO:** This record defines a line from the current drawing position up to, but not including, the specified point. It resets the current drawing position to the specified point.

**EMR\_ARCTO:** This record defines an elliptical arc. It resets the current position to the endpoint of the arc.

**EMR\_POLYDRAW:** This record defines a set of line segments and Bezier curves.

**EMR\_SETARCDIRECTION:** This record defines the drawing direction to be used for arc and rectangle operations.

**EMR\_SETMITERLIMIT:** This record defines the limit for the length of miter joins.

**EMR\_BEGINPATH:** This record opens a [**path bracket**](#gt_b89ea7da-c0bf-4c45-9f3e-fc430d7ab601) for specifying the current [**path**](#gt_2cd71385-2d9c-4ab8-bf4a-7b258816d613).

**EMR\_ENDPATH:** This record closes an open path bracket and selects the path into the playback device context.

**EMR\_CLOSEFIGURE:** This record closes an open figure in a path.

**EMR\_FILLPATH:** This record closes any open figures in the current path bracket and fills its interior by using the current brush and polygon-filling mode.

**EMR\_STROKEANDFILLPATH:** This record closes any open figures in a path, strokes the outline of the path by using the current pen, and fills its interior by using the current brush.

**EMR\_STROKEPATH:** This record renders the specified path by using the current pen.

**EMR\_FLATTENPATH:** This record turns each curve in the path into a sequence of lines.

**EMR\_WIDENPATH:** This record redefines the current path bracket as the area that would be painted if the path were stroked using the current pen.

**EMR\_SELECTCLIPPATH:** This record specifies a clipping region as the current clipping region combined with the current path bracket, using the specified mode.

**EMR\_ABORTPATH:** This record aborts a path bracket or discards the path from a closed path bracket.

**EMR\_COMMENT:** This record specifies arbitrary private data.

**EMR\_FILLRGN:** This record fills the specified region by using the specified brush.

**EMR\_FRAMERGN:** This record draws a border around the specified region using the specified brush.

**EMR\_INVERTRGN:** This record inverts the colors in the specified region.

**EMR\_PAINTRGN:** This record paints the specified region by using the current brush.

**EMR\_EXTSELECTCLIPRGN:** This record combines the specified region with the current clipping region, using the specified mode.

**EMR\_BITBLT:** This record specifies a block transfer of pixels from a source bitmap to a destination rectangle, optionally in combination with a brush pattern, according to a specified raster operation.

**EMR\_STRETCHBLT:** This record specifies a block transfer of pixels from a source bitmap to a destination rectangle, optionally in combination with a brush pattern, according to a specified raster operation, stretching or compressing the output to fit the dimensions of the destination, if necessary.

**EMR\_MASKBLT:** This record specifies a block transfer of pixels from a source bitmap to a destination rectangle, optionally in combination with a brush pattern and with the application of a color mask bitmap, according to specified foreground and background raster operations.

**EMR\_PLGBLT:** This record specifies a block transfer of pixels from a source bitmap to a destination parallelogram, with the application of a color mask bitmap.

**EMR\_SETDIBITSTODEVICE:** This record specifies a block transfer of pixels from specified scanlines of a source bitmap to a destination rectangle.

**EMR\_STRETCHDIBITS:** This record specifies a block transfer of pixels from a source bitmap to a destination rectangle, optionally in combination with a brush pattern, according to a specified raster operation, stretching or compressing the output to fit the dimensions of the destination, if necessary.

**EMR\_EXTCREATEFONTINDIRECTW:** This record defines a logical font that has the specified characteristics. The font can subsequently be selected as the current font.

**EMR\_EXTTEXTOUTA:** This record draws an [**ASCII**](#gt_79fa85ca-ac61-467c-b819-e97dc1a7a599) text string using the current font and text colors.

**EMR\_EXTTEXTOUTW:** This record draws a [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) text string using the current font and text colors.

**EMR\_POLYBEZIER16:** This record defines one or more Bezier curves. The curves are drawn using the current pen.

**EMR\_POLYGON16:** This record defines a polygon consisting of two or more vertexes connected by straight lines. The polygon is outlined by using the current pen and filled by using the current brush and polygon fill mode. The polygon is closed automatically by drawing a line from the last vertex to the first.

**EMR\_POLYLINE16:** This record defines a series of line segments by connecting the points in the specified array.

**EMR\_POLYBEZIERTO16:** This record defines one or more Bezier curves based on the current position.

**EMR\_POLYLINETO16:** This record defines one or more straight lines based upon the current position. A line is drawn from the current position to the first point specified by the **Points** field by using the current pen. For each additional line, drawing is performed from the ending point of the previous line to the next point specified by **Points**.

**EMR\_POLYPOLYLINE16:** This record defines multiple series of connected line segments.

**EMR\_POLYPOLYGON16:** This record defines a series of closed polygons. Each polygon is outlined by using the current pen and filled by using the current brush and polygon fill mode. The polygons specified by this record can overlap.

**EMR\_POLYDRAW16:** This record defines a set of line segments and Bezier curves.

**EMR\_CREATEMONOBRUSH:** This record defines a logical brush with the specified bitmap pattern. The bitmap can be a [**device-independent bitmap (DIB)**](#gt_3a6934d5-cc59-4ab9-a54e-e062a09a4091) section bitmap or it can be a device-dependent bitmap.

**EMR\_CREATEDIBPATTERNBRUSHPT:** This record defines a logical brush that has the pattern specified by the DIB.

**EMR\_EXTCREATEPEN:** This record defines an extended logical pen (section [2.2.20](#Section_5b67b3eeea004f809b732959804381be)) that has the specified style, width, color, and brush attributes.

**EMR\_POLYTEXTOUTA:** This record draws one or more ASCII text strings using the current font and text colors.

**Note:** [EMR\_POLYTEXTOUTA](#Section_d8b7ac2176b84f9aa7cc05f9d2d5627e) SHOULD be emulated with a series of [EMR\_EXTTEXTOUTW](#Section_a59a79ac328e492da34de02727af6edf) records, one per string.[<3>](#Appendix_A_3" \o "Product behavior note 3)

**EMR\_POLYTEXTOUTW:** This record draws one or more Unicode text strings using the current font and text colors.

**Note:** [EMR\_POLYTEXTOUTW](#Section_381015eb29a7477493ada210697f5972) SHOULD be emulated with a series of EMR\_EXTTEXTOUTW records, one per string.[<4>](#Appendix_A_4" \o "Product behavior note 4)

**EMR\_SETICMMODE:** This record specifies the mode of [**Image Color Management (ICM)**](#gt_fcd5ee55-57ff-45d6-b51e-2ccba47329e3) for graphics operations.[<5>](#Appendix_A_5" \o "Product behavior note 5)

**EMR\_CREATECOLORSPACE:** This record creates a logical [**color space**](#gt_5d3fb5ea-c686-4d3b-b0ff-aef33fe1aee5) object from a [**color profile**](#gt_cffe5dfa-5408-4302-8ab0-400cb94f5e32) with a name consisting of ASCII characters.[<6>](#Appendix_A_6" \o "Product behavior note 6)

**EMR\_SETCOLORSPACE:** This record defines the current logical color space object for graphics operations.[<7>](#Appendix_A_7" \o "Product behavior note 7)

**EMR\_DELETECOLORSPACE:** This record deletes a logical color space object.[<8>](#Appendix_A_8" \o "Product behavior note 8)

**Note:** An [EMR\_DELETEOBJECT](#Section_6f0f12a3111a478b8251a9505168f9a9) record SHOULD be used instead of [EMR\_DELETECOLORSPACE](#Section_5d137387d79a4bc89a4d38291320e148) to delete a logical color space object.[<9>](#Appendix_A_9" \o "Product behavior note 9)

**EMR\_GLSRECORD:** This record specifies an [**OpenGL**](#gt_216becb9-a800-4a44-8206-2ec2231c9c67) function.[<10>](#Appendix_A_10" \o "Product behavior note 10)

**EMR\_GLSBOUNDEDRECORD:** This record specifies an OpenGL function with a bounding rectangle for output.[<11>](#Appendix_A_11" \o "Product behavior note 11)

**EMR\_PIXELFORMAT:** This record specifies the pixel format to use for graphics operations.[<12>](#Appendix_A_12" \o "Product behavior note 12)

**EMR\_DRAWESCAPE:** This record passes arbitrary information to the driver. The intent is that the information results in drawing being done.

**EMR\_EXTESCAPE:** This record passes arbitrary information to the driver. The intent is that the information does not result in drawing being done.

**EMR\_SMALLTEXTOUT:** This record outputs a string.

**EMR\_FORCEUFIMAPPING:** This record forces the font mapper to match fonts based on their **UniversalFontId** in preference to their **LogFont** information.

**EMR\_NAMEDESCAPE:** This record passes arbitrary information to the given named driver.

**EMR\_COLORCORRECTPALETTE:** This record specifies how to correct the entries of a logical palette object using [**Windows Color System (WCS)**](#gt_e4f9c4a4-c195-43d0-b189-f9727ec8a2b3) 1.0 values.[<13>](#Appendix_A_13" \o "Product behavior note 13)

**EMR\_SETICMPROFILEA:** This record specifies a color profile in a file with a name consisting of ASCII characters, for graphics output.[<14>](#Appendix_A_14" \o "Product behavior note 14)

**EMR\_SETICMPROFILEW:** This record specifies a color profile in a file with a name consisting of Unicode characters, for graphics output.[<15>](#Appendix_A_15" \o "Product behavior note 15)

**EMR\_ALPHABLEND:** This record specifies a block transfer of pixels from a source bitmap to a destination rectangle, including [**alpha transparency**](#gt_134d5688-b046-4a5b-a74e-c0c11fc04eb3) data, according to a specified blending operation.[<16>](#Appendix_A_16" \o "Product behavior note 16)

**EMR\_SETLAYOUT:** This record specifies the order in which text and graphics are drawn.[<17>](#Appendix_A_17" \o "Product behavior note 17)

**EMR\_TRANSPARENTBLT:** This record specifies a block transfer of pixels from a source bitmap to a destination rectangle, treating a specified color as transparent, stretching or compressing the output to fit the dimensions of the destination, if necessary.[<18>](#Appendix_A_18" \o "Product behavior note 18)

**EMR\_GRADIENTFILL:** This record specifies filling rectangles or triangles with gradients of color.[<19>](#Appendix_A_19" \o "Product behavior note 19)

**EMR\_SETLINKEDUFIS:** This record sets the **UniversalFontIds** (section [2.2.27](#Section_34502dab22b44855b9b288f00a4a0c52)) of linked fonts to use during character lookup.

**EMR\_SETTEXTJUSTIFICATION:** This record specifies the amount of extra space to add to break characters for justification purposes.[<20>](#Appendix_A_20" \o "Product behavior note 20)

**EMR\_COLORMATCHTOTARGETW:** This record specifies whether to perform [**color matching**](#gt_a3322d29-7dc5-4696-b17a-55311db1cb33) with a color profile that is specified in a file with a name consisting of Unicode characters.[<21>](#Appendix_A_21" \o "Product behavior note 21)

**EMR\_CREATECOLORSPACEW:** This record creates a logical color space object from a color profile with a name consisting of Unicode characters.[<22>](#Appendix_A_22" \o "Product behavior note 22)

### ArcDirection Enumeration

The **ArcDirection** enumeration is used in setting the drawing direction for arcs and rectangles.

1. typedef enum
2. {
3. AD\_COUNTERCLOCKWISE = 0x00000001,
4. AD\_CLOCKWISE = 0x00000002
5. } ArcDirection;

**AD\_COUNTERCLOCKWISE:** Figures drawn counterclockwise.

**AD\_CLOCKWISE:** Figures drawn clockwise.

### ArmStyle Enumeration

The ArmStyle enumeration defines values for one of the characteristics in the [**PANOSE**](#gt_8290d13f-8670-4b8e-ae6e-1ebb49319792) system for classifying [**typefaces**](#gt_2fff9b0b-4bc1-4d05-81ef-8dde59fd9789).

1. typedef enum
2. {
3. PAN\_ANY = 0x00,
4. PAN\_NO\_FIT = 0x01,
5. PAN\_STRAIGHT\_ARMS\_HORZ = 0x02,
6. PAN\_STRAIGHT\_ARMS\_WEDGE = 0x03,
7. PAN\_STRAIGHT\_ARMS\_VERT = 0x04,
8. PAN\_STRAIGHT\_ARMS\_SINGLE\_SERIF = 0x05,
9. PAN\_STRAIGHT\_ARMS\_DOUBLE\_SERIF = 0x06,
10. PAN\_BENT\_ARMS\_HORZ = 0x07,
11. PAN\_BENT\_ARMS\_WEDGE = 0x08,
12. PAN\_BENT\_ARMS\_VERT = 0x09,
13. PAN\_BENT\_ARMS\_SINGLE\_SERIF = 0x0A,
14. PAN\_BENT\_ARMS\_DOUBLE\_SERIF = 0x0B
15. } ArmStyle;

**PAN\_ANY:** Any.

**PAN\_NO\_FIT:** No fit.

**PAN\_STRAIGHT\_ARMS\_HORZ:** Straight arms/horizontal.

**PAN\_STRAIGHT\_ARMS\_WEDGE:** Straight arms/wedge.

**PAN\_STRAIGHT\_ARMS\_VERT:** Straight arms/vertical.

**PAN\_STRAIGHT\_ARMS\_SINGLE\_SERIF:** Straight arms/single-serif.

**PAN\_STRAIGHT\_ARMS\_DOUBLE\_SERIF:** Straight arms/double-serif.

**PAN\_BENT\_ARMS\_HORZ:** Nonstraight arms/horizontal.

**PAN\_BENT\_ARMS\_WEDGE:** Nonstraight arms/wedge.

**PAN\_BENT\_ARMS\_VERT:** Nonstraight arms/vertical.

**PAN\_BENT\_ARMS\_SINGLE\_SERIF:** Nonstraight arms/single-serif.

**PAN\_BENT\_ARMS\_DOUBLE\_SERIF:** Nonstraight arms/double-serif.

### BackgroundMode Enumeration

The **BackgroundMode** enumeration is used to specify the background mode to be used with text, hatched brushes, and pen styles that are not solid. The background mode determines how to combine the background with foreground text, hatched brushes, and pen styles that are not solid lines.

1. typedef enum
2. {
3. TRANSPARENT = 0x0001,
4. OPAQUE = 0x0002
5. } BackgroundMode;

**TRANSPARENT:** Background remains untouched.

**OPAQUE:** Background is filled with the current background color before the text, hatched brush, or pen is drawn.

### ColorAdjustment Enumeration

The ColorAdjustment enumeration is used to specify how the output image is prepared when the stretch mode is **HALFTONE**.

1. typedef enum
2. {
3. CA\_NEGATIVE = 0x0001,
4. CA\_LOG\_FILTER = 0x0002
5. } ColorAdjustment;

**CA\_NEGATIVE:** Specifies that the negative of the original image SHOULD be displayed.

**CA\_LOG\_FILTER:** Specifies that a logarithmic process SHOULD be applied to the final density of the output colors. This will increase the color contrast when the luminance is low.

### ColorMatchToTarget Enumeration

The **ColorMatchToTarget** enumeration is used to determine whether a [**color profile**](#gt_cffe5dfa-5408-4302-8ab0-400cb94f5e32) has been embedded in the [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7).

1. typedef enum
2. {
3. COLORMATCHTOTARGET\_NOTEMBEDDED = 0x00000000,
4. COLORMATCHTOTARGET\_EMBEDDED = 0x00000001
5. } ColorMatchToTarget;

**COLORMATCHTOTARGET\_NOTEMBEDDED:** Indicates that a color profile has not been embedded in the metafile.

**COLORMATCHTOTARGET\_EMBEDDED:** Indicates that a color profile has been embedded in the metafile.

### ColorSpace Enumeration

The **ColorSpace** enumeration is used to specify when to turn [**color proofing**](#gt_f00dd816-fa37-4de3-b30c-1644cbbd45a4) on and off, and when to delete [**transforms**](#gt_fdca4a96-8e04-4dc0-acad-1d01e1384a5b).

1. typedef enum
2. {
3. CS\_ENABLE = 0x00000001,
4. CS\_DISABLE = 0x00000002,
5. CS\_DELETE\_TRANSFORM = 0x00000003
6. } ColorSpace;

**CS\_ENABLE:** Maps colors to the target device's [**color gamut**](#gt_cc6ddf19-b7de-4820-8b48-ecd73c45eef2). This enables color proofing. All subsequent draw commands render colors as they would appear on the target device.

**CS\_DISABLE:** Disables color proofing.

**CS\_DELETE\_TRANSFORM:** If color management is enabled for the target profile, disables it and deletes the current color transform.

### Contrast Enumeration

The Contrast enumeration defines values for one of the characteristics in the [**PANOSE**](#gt_8290d13f-8670-4b8e-ae6e-1ebb49319792) system for classifying [**typefaces**](#gt_2fff9b0b-4bc1-4d05-81ef-8dde59fd9789).

1. typedef enum
2. {
3. PAN\_ANY = 0x00,
4. PAN\_NO\_FIT = 0x01,
5. PAN\_CONTRAST\_NONE = 0x02,
6. PAN\_CONTRAST\_VERY\_LOW = 0x03,
7. PAN\_CONTRAST\_LOW = 0x04,
8. PAN\_CONTRAST\_MEDIUM\_LOW = 0x05,
9. PAN\_CONTRAST\_MEDIUM = 0x06,
10. PAN\_CONTRAST\_MEDIUM\_HIGH = 0x07,
11. PAN\_CONTRAST\_HIGH = 0x08,
12. PAN\_CONTRAST\_VERY\_HIGH = 0x09
13. } Contrast;

**PAN\_ANY:** Any.

**PAN\_NO\_FIT:** No fit.

**PAN\_CONTRAST\_NONE:** None.

**PAN\_CONTRAST\_VERY\_LOW:** Very low.

**PAN\_CONTRAST\_LOW:** Low.

**PAN\_CONTRAST\_MEDIUM\_LOW:** Medium low.

**PAN\_CONTRAST\_MEDIUM:** Medium.

**PAN\_CONTRAST\_MEDIUM\_HIGH:** Medium high.

**PAN\_CONTRAST\_HIGH:** High.

**PAN\_CONTRAST\_VERY\_HIGH:** Very high.

### DIBColors Enumeration

The **DIBColors** enumeration defines how to interpret the values in the color table of a [**DIB**](#gt_3a6934d5-cc59-4ab9-a54e-e062a09a4091).

1. typedef enum
2. {
3. DIB\_RGB\_COLORS = 0x00,
4. DIB\_PAL\_COLORS = 0x01,
5. DIB\_PAL\_INDICES = 0x02
6. } DIBColors;

**DIB\_RGB\_COLORS:** The color table contains literal [**RGB**](#gt_2c716d3a-e60b-4e52-bbb0-2fdeb298003b) values.

**DIB\_PAL\_COLORS:** The color table consists of an array of 16-bit indexes into the LogPalette object (section [2.2.17](#Section_758f047d765e424c9204a833b7b4e527)) that is currently defined in the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45).

**DIB\_PAL\_INDICES:** No color table exists. The pixels in the DIB are indices into the current logical palette in the playback device context.

DIBs are specified by DeviceIndependentBitmap objects ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.9).

### EmrComment Enumeration

The **EmrComment** enumeration defines the types of data that a public comment record can contain, as specified in section [2.3.3.4](#Section_39c91f22e4cc4e82a6050137caa8457c).

1. typedef enum
2. {
3. EMR\_COMMENT\_WINDOWS\_METAFILE = 0x80000001,
4. EMR\_COMMENT\_BEGINGROUP = 0x00000002,
5. EMR\_COMMENT\_ENDGROUP = 0x00000003,
6. EMR\_COMMENT\_MULTIFORMATS = 0x40000004,
7. EMR\_COMMENT\_UNICODE\_STRING = 0x00000040,
8. EMR\_COMMENT\_UNICODE\_END = 0x00000080
9. } EmrComment;

**EMR\_COMMENT\_WINDOWS\_METAFILE:** This comment record contains a specification of an image in [**WMF**](#gt_48849cf6-d55c-47e5-b041-13b65854de2b) [[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2).

**EMR\_COMMENT\_BEGINGROUP:** This comment record identifies the beginning of a group of drawing records.

**EMR\_COMMENT\_ENDGROUP:** This comment record identifies the end of a group of drawing records.

**EMR\_COMMENT\_MULTIFORMATS:** This comment record allows multiple definitions of an image to be included in the [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7). Using this comment, for example, an application can include encapsulated [**PostScript**](#gt_31971c63-f974-47de-8de9-6b76d093890d) text as well as an [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) definition of an image.

**EMR\_COMMENT\_UNICODE\_STRING:** This comment record is reserved and MUST NOT be used.

**EMR\_COMMENT\_UNICODE\_END:** This comment record is reserved and MUST NOT be used.

### ExtTextOutOptions Enumeration

The ExtTextOutOptions enumeration specifies parameters that control various aspects of the output of text by [EMR\_SMALLTEXTOUT (section 2.3.5.37)](#Section_20eee81d0bd442d1a624860adfe62358) records and in [EmrText](#Section_dd585d0a5d7c4034963a1141af836972) objects.

1. typedef enum
2. {
3. ETO\_OPAQUE = 0x00000002,
4. ETO\_CLIPPED = 0x00000004,
5. ETO\_GLYPH\_INDEX = 0x00000010,
6. ETO\_RTLREADING = 0x00000080,
7. ETO\_NO\_RECT = 0x00000100,
8. ETO\_SMALL\_CHARS = 0x00000200,
9. ETO\_NUMERICSLOCAL = 0x00000400,
10. ETO\_NUMERICSLATIN = 0x00000800,
11. ETO\_IGNORELANGUAGE = 0x00001000,
12. ETO\_PDY = 0x00002000,
13. ETO\_REVERSE\_INDEX\_MAP = 0x00010000
14. } ExtTextOutOptions;

**ETO\_OPAQUE:** This bit indicates that the current background color SHOULD be used to fill the rectangle.

**ETO\_CLIPPED:** This bit indicates that the text SHOULD be clipped to the rectangle.

**ETO\_GLYPH\_INDEX:** This bit indicates that the codes for characters in an output text string are indexes of the character glyphs in a [**TrueType**](#gt_855615c9-5378-459f-8d58-20f640afed6e) font. Glyph indexes are font-specific, so to display the correct characters on playback, the font that is used MUST be identical to the font used to generate the indexes.[<23>](#Appendix_A_23" \o "Product behavior note 23)

**ETO\_RTLREADING:** This bit indicates that the text MUST be laid out in right-to-left reading order, instead of the default left-to-right order. This SHOULD be applied only when the font selected into the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45) is either Hebrew or Arabic.[<24>](#Appendix_A_24" \o "Product behavior note 24)

**ETO\_NO\_RECT:** This bit indicates that the record does not specify a bounding rectangle for the text output.

**ETO\_SMALL\_CHARS:** This bit indicates that the codes for characters in an output text string are 8 bits, derived from the low bytes of [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) [**UTF16-LE**](#gt_f25550c9-f84f-4eb2-8156-14794a7e3059) character codes, in which the high byte is assumed to be 0.

**ETO\_NUMERICSLOCAL:** This bit indicates that to display numbers, digits appropriate to the locale SHOULD be used.[<25>](#Appendix_A_25" \o "Product behavior note 25)

**ETO\_NUMERICSLATIN:** This bit indicates that to display numbers, European digits SHOULD be used.[<26>](#Appendix_A_26" \o "Product behavior note 26)

**ETO\_IGNORELANGUAGE:** This bit indicates that no special operating system processing for glyph placement is performed on right-to-left strings; that is, all glyph positioning SHOULD be taken care of by drawing and state records in the [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7).[<27>](#Appendix_A_27" \o "Product behavior note 27)

**ETO\_PDY:** This bit indicates that both horizontal and vertical character displacement values SHOULD be provided.[<28>](#Appendix_A_28" \o "Product behavior note 28)

**ETO\_REVERSE\_INDEX\_MAP:** This bit is reserved and SHOULD NOT be used.[<29>](#Appendix_A_29" \o "Product behavior note 29)

### FamilyType Enumeration

The FamilyType enumeration defines values for one of the characteristics in the [**PANOSE**](#gt_8290d13f-8670-4b8e-ae6e-1ebb49319792) system for classifying [**typefaces**](#gt_2fff9b0b-4bc1-4d05-81ef-8dde59fd9789).

1. typedef enum
2. {
3. PAN\_ANY = 0x00,
4. PAN\_NO\_FIT = 0x01,
5. PAN\_FAMILY\_TEXT\_DISPLAY = 0x02,
6. PAN\_FAMILY\_SCRIPT = 0x03,
7. PAN\_FAMILY\_DECORATIVE = 0x04,
8. PAN\_FAMILY\_PICTORIAL = 0x05
9. } FamilyType;

**PAN\_ANY:** Any.

**PAN\_NO\_FIT:** No fit.

**PAN\_FAMILY\_TEXT\_DISPLAY:** Text and display.

**PAN\_FAMILY\_SCRIPT:** Script.

**PAN\_FAMILY\_DECORATIVE:** Decorative.

**PAN\_FAMILY\_PICTORIAL:** Pictorial.

### FloodFill Enumeration

The FloodFill enumeration defines values that specify how to determine the area for a flood fill operation.

1. typedef enum
2. {
3. FLOODFILLBORDER = 0x00000000,
4. FLOODFILLSURFACE = 0x00000001
5. } FloodFill;

**FLOODFILLBORDER:** The fill area is bounded by a specific color.

**FLOODFILLSURFACE:** The fill area is defined by a specific color. Filling continues outward in all directions as long as the color is encountered. This style is useful for filling areas with multicolored boundaries.

### FormatSignature Enumeration

The **FormatSignature** enumeration defines values that are used to identify the format of embedded data in [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) [**metafiles**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7).

1. typedef enum
2. {
3. ENHMETA\_SIGNATURE = 0x464D4520,
4. EPS\_SIGNATURE = 0x46535045
5. } FormatSignature;

**ENHMETA\_SIGNATURE:** The sequence of [**ASCII**](#gt_79fa85ca-ac61-467c-b819-e97dc1a7a599) characters "FME ", which denotes EMF data. The reverse of the string is " EMF".

**Note:** The space character in the string is significant and MUST be present.

This signature is used in the following structures:

* EMR\_HEADER records (section [2.3.4.2](#Section_de081cd7351f4cc2830bd03fb55e89ab)) to identify the EMF metafile
* The EmrFormat object (section [2.2.4](#Section_1bd63563ce674528bf285a8852ff0270)) in EMR\_COMMENT\_MULTIFORMATS records (section [2.3.3.4.3](#Section_4bc4608eca82426dac0e5ded2ad3fc4e)), to specify embedded EMF records.

**EPS\_SIGNATURE:** The value of this member is the sequence of ASCII characters "FSPE", which denotes [**encapsulated PostScript (EPS)**](#gt_ba716063-a4a2-4e6e-a657-7ba9fde7b93f) data. The reverse of the string is "EPSF".

This signature is used in EmrFormat objects to specify embedded [**PostScript**](#gt_31971c63-f974-47de-8de9-6b76d093890d) data in the EpsData object (section [2.2.6](#Section_a624b402f7aa4cb6b289612c9775f37e)) in EMR\_COMMENT\_MULTIFORMATS records.

### GradientFill Enumeration

The **GradientFill** enumeration defines the modes for gradient fill operations.

1. typedef enum
2. {
3. GRADIENT\_FILL\_RECT\_H = 0x00000000,
4. GRADIENT\_FILL\_RECT\_V = 0x00000001,
5. GRADIENT\_FILL\_TRIANGLE = 0x00000002
6. } GradientFill;

**GRADIENT\_FILL\_RECT\_H:** Color interpolation along a gradient from the left to the right edges of a rectangle.

**GRADIENT\_FILL\_RECT\_V:** Color interpolation along a gradient from the top to the bottom edges of a rectangle.

**GRADIENT\_FILL\_TRIANGLE:** Color interpolation between vertexes of a triangle.

### GraphicsMode Enumeration

The **GraphicsMode** enumeration is used to specify how to interpret shape data such as rectangle coordinates.

1. typedef enum
2. {
3. GM\_COMPATIBLE = 0x00000001,
4. GM\_ADVANCED = 0x00000002
5. } GraphicsMode;

**GM\_COMPATIBLE:** [**TrueType**](#gt_855615c9-5378-459f-8d58-20f640afed6e) text MUST be written from left to right and right side up, even if the rest of the graphics are [**rotated**](#gt_e833af67-6604-4d48-b5a2-84ea0cbbbdbb) about the x-axis or y-axis because of the current world-to-device [**transform**](#gt_fdca4a96-8e04-4dc0-acad-1d01e1384a5b). Only the height of the text SHOULD be scaled.[<30>](#Appendix_A_30" \o "Product behavior note 30)

Arcs MUST be drawn using the current arc direction, but they MUST NOT reflect the current world-to-device transform, which might require a [**rotation**](#gt_5cdd5e5b-fe3d-46e6-9019-50483463a76a) along the x-axis or y-axis.

The world-to-device transform is modified by changing the window and viewport extents and origins, using the EMR\_SETWINDOWEXTEX (section [2.3.11.30](#Section_1952148360b549b8b58c4b5d1b6cf839)) and EMR\_SETVIEWPORTEXTEX (section [2.3.11.28](#Section_a71486a39ebd43e3819f5318a58b70d5)) records, and the EMR\_SETWINDOWORGEX (section [2.3.11.31](#Section_952803a93d5c4c83a258558bc1123bdb)) and EMR\_SETVIEWPORTORGEX (section [2.3.11.29](#Section_b89489473f8d4adda4b8e572e8f90506)) records, respectively.

The world-to-device transform can be changed by EMR\_MODIFYWORLDTRANSFORM (section [2.3.12.1](#Section_c70b85e58c31418fa7b8349e417e0f76)) and EMR\_SETWORLDTRANSFORM (section [2.3.12.2](#Section_985724c04db148f0b34667288b3288cb)) records.

In **GM\_COMPATIBLE** graphics mode, bottom and rightmost edges MUST be excluded when rectangles are drawn.

**GM\_ADVANCED:** TrueType text output SHOULD[<31>](#Appendix_A_31" \o "Product behavior note 31) fully conform to the current world-to-device transform.

Arcs MUST be drawn in the counterclockwise direction in [**world space**](#gt_baadb707-1343-413b-9b36-2496b06e9bd7); however, both arc control points and the arcs themselves MUST reflect the current world-to-device transform.

The world-to-device transform can be modified directly by EMR\_MODIFYWORLDTRANSFORM and EMR\_SETWORLDTRANSFORM records, or indirectly by changing the window and viewport extents and origins, using the EMR\_SETWINDOWEXTEX and EMR\_SETVIEWPORTEXTEX records, and the EMR\_SETWINDOWORGEX and EMR\_SETVIEWPORTORGEX records, respectively.

In **GM\_ADVANCED** graphics mode, bottom and rightmost edges MUST be included when rectangles are drawn.

### HatchStyle Enumeration

The **HatchStyle** enumeration is an extension to the [**WMF**](#gt_48849cf6-d55c-47e5-b041-13b65854de2b) **HatchStyle** enumeration ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.1.1.12).

1. typedef enum
2. {
3. HS\_SOLIDCLR = 0x0006,
4. HS\_DITHEREDCLR = 0x0007,
5. HS\_SOLIDTEXTCLR = 0x0008,
6. HS\_DITHEREDTEXTCLR = 0x0009,
7. HS\_SOLIDBKCLR = 0x000A,
8. HS\_DITHEREDBKCLR = 0x000B
9. } HatchStyle;

**HS\_SOLIDCLR:** The hatch is not a pattern, but is a solid color.

**HS\_DITHEREDCLR:** The hatch is not a pattern, but is a dithered color.

**HS\_SOLIDTEXTCLR:** The hatch is not a pattern, but is a solid color, defined by the current text (foreground) color.

**HS\_DITHEREDTEXTCLR:** The hatch is not a pattern, but is a dithered color, defined by the current text (foreground) color.

**HS\_SOLIDBKCLR:** The hatch is not a pattern, but is a solid color, defined by the current background color.

**HS\_DITHEREDBKCLR:** The hatch is not a pattern, but is a dithered color, defined by the current background color.

### ICMMode Enumeration

The **ICMMode** enumeration defines values that specify when to turn on and off [**Image Color Management (ICM)**](#gt_fcd5ee55-57ff-45d6-b51e-2ccba47329e3).[<32>](#Appendix_A_32" \o "Product behavior note 32)

1. typedef enum
2. {
3. ICM\_OFF = 0x01,
4. ICM\_ON = 0x02,
5. ICM\_QUERY = 0x03,
6. ICM\_DONE\_OUTSIDEDC = 0x04
7. } ICMMode;

**ICM\_OFF:** Turns off ICM; turns on old-style [**color correction**](#gt_0f6a2bd0-2201-4ce8-9b84-bd5a64a79931) of [**halftones**](#gt_476862c3-a702-4d6c-8f3b-a42dc49cce79).

**ICM\_ON:** Turns on ICM; turns off old-style color correction of halftones.

**ICM\_QUERY:** Queries the current state of color management.

**ICM\_DONE\_OUTSIDEDC:** Turns off both ICM and old-style color correction of halftones.

### Illuminant Enumeration

The **Illuminant** enumeration defines values that specify the illuminant value of an image, which determines the standard light source under which the image is viewed so that the color can be adjusted appropriately.

1. typedef enum
2. {
3. ILLUMINANT\_DEVICE\_DEFAULT = 0x00,
4. ILLUMINANT\_TUNGSTEN = 0x01,
5. ILLUMINANT\_B = 0x02,
6. ILLUMINANT\_DAYLIGHT = 0x03,
7. ILLUMINANT\_D50 = 0x04,
8. ILLUMINANT\_D55 = 0x05,
9. ILLUMINANT\_D65 = 0x06,
10. ILLUMINANT\_D75 = 0x07,
11. ILLUMINANT\_FLUORESCENT = 0x08
12. } Illuminant;

**ILLUMINANT\_DEVICE\_DEFAULT:** Device's default. Standard used by output devices.

**ILLUMINANT\_TUNGSTEN:** Tungsten lamp.

**ILLUMINANT\_B:** Noon sunlight.

**ILLUMINANT\_DAYLIGHT:** Daylight.

**ILLUMINANT\_D50:** Normal print.

**ILLUMINANT\_D55:** Bond paper print.

**ILLUMINANT\_D65:** Standard daylight. Standard for CRTs and pictures.

**ILLUMINANT\_D75:** Northern daylight.

**ILLUMINANT\_FLUORESCENT:** Cool white lamp.

### Letterform Enumeration

The **Letterform** enumeration defines values for one of the characteristics in the [**PANOSE**](#gt_8290d13f-8670-4b8e-ae6e-1ebb49319792) system for classifying [**typefaces**](#gt_2fff9b0b-4bc1-4d05-81ef-8dde59fd9789).

1. typedef enum
2. {
3. PAN\_ANY = 0x00,
4. PAN\_NO\_FIT = 0x01,
5. PAN\_LETT\_NORMAL\_CONTACT = 0x02,
6. PAN\_LETT\_NORMAL\_WEIGHTED = 0x03,
7. PAN\_LETT\_NORMAL\_BOXED = 0x04,
8. PAN\_LETT\_NORMAL\_FLATTENED = 0x05,
9. PAN\_LETT\_NORMAL\_ROUNDED = 0x06,
10. PAN\_LETT\_NORMAL\_OFF\_CENTER = 0x07,
11. PAN\_LETT\_NORMAL\_SQUARE = 0x08,
12. PAN\_LETT\_OBLIQUE\_CONTACT = 0x09,
13. PAN\_LETT\_OBLIQUE\_WEIGHTED = 0x0A,
14. PAN\_LETT\_OBLIQUE\_BOXED = 0x0B,
15. PAN\_LETT\_OBLIQUE\_FLATTENED = 0x0C,
16. PAN\_LETT\_OBLIQUE\_ROUNDED = 0x0D,
17. PAN\_LETT\_OBLIQUE\_OFF\_CENTER = 0x0E,
18. PAN\_LETT\_OBLIQUE\_SQUARE = 0x0F
19. } Letterform;

**PAN\_ANY:** Any.

**PAN\_NO\_FIT:** No fit.

**PAN\_LETT\_NORMAL\_CONTACT:** Normal/contact.

**PAN\_LETT\_NORMAL\_WEIGHTED:** Normal/weighted.

**PAN\_LETT\_NORMAL\_BOXED:** Normal/boxed.

**PAN\_LETT\_NORMAL\_FLATTENED:** Normal/flattened.

**PAN\_LETT\_NORMAL\_ROUNDED:** Normal/rounded.

**PAN\_LETT\_NORMAL\_OFF\_CENTER:** Normal/off center.

**PAN\_LETT\_NORMAL\_SQUARE:** Normal/square

**PAN\_LETT\_OBLIQUE\_CONTACT:** Oblique/contact.

**PAN\_LETT\_OBLIQUE\_WEIGHTED:** Oblique/weighted.

**PAN\_LETT\_OBLIQUE\_BOXED:** Oblique/boxed.

**PAN\_LETT\_OBLIQUE\_FLATTENED:** Oblique/flattened.

**PAN\_LETT\_OBLIQUE\_ROUNDED:** Oblique/rounded.

**PAN\_LETT\_OBLIQUE\_OFF\_CENTER:** Oblique/off center.

**PAN\_LETT\_OBLIQUE\_SQUARE:** Oblique/square.

### MapMode Enumeration

The **MapMode** enumeration is used to define the unit of measure for transforming [**page space**](#gt_f530f123-858a-4af7-b626-adea0aeaa75a) units into [**device space**](#gt_25d586c7-f440-45c6-b965-5c025bea9fc0) units and for defining the orientation of the drawing axes.

1. typedef enum
2. {
3. MM\_TEXT = 0x01,
4. MM\_LOMETRIC = 0x02,
5. MM\_HIMETRIC = 0x03,
6. MM\_LOENGLISH = 0x04,
7. MM\_HIENGLISH = 0x05,
8. MM\_TWIPS = 0x06,
9. MM\_ISOTROPIC = 0x07,
10. MM\_ANISOTROPIC = 0x08
11. } MapMode;

**MM\_TEXT:** Each logical unit is mapped to one device pixel. Positive x is to the right; positive y is down.

**MM\_LOMETRIC:** Each logical unit is mapped to 0.1 millimeter. Positive x is to the right; positive y is up.

**MM\_HIMETRIC:** Each logical unit is mapped to 0.01 millimeter. Positive x is to the right; positive y is up.

**MM\_LOENGLISH:** Each logical unit is mapped to 0.01 inch. Positive x is to the right; positive y is up.

**MM\_HIENGLISH:** Each logical unit is mapped to 0.001 inch. Positive x is to the right; positive y is up.

**MM\_TWIPS:** Each logical unit is mapped to one-twentieth of a printer's point (1/1440 inch, also called a "twip"). Positive x is to the right; positive y is up.

**MM\_ISOTROPIC:** Logical units are [**isotropic**](#gt_716e3252-cd28-451e-be10-bde8793f0534); that is, they are mapped to arbitrary units with equally scaled axes. Thus, one unit along the x-axis is equal to one unit along the y-axis. The EMR\_SETWINDOWEXTEX (section [2.3.11.30](#Section_1952148360b549b8b58c4b5d1b6cf839)) and EMR\_SETVIEWPORTEXTEX (section [2.3.11.28](#Section_a71486a39ebd43e3819f5318a58b70d5)) records are used to specify the units and the orientation of the axes.

Adjustments MUST be made as necessary to ensure that the x and y units remain the same size. For example, when the window extent is set, the viewport MUST be adjusted to keep the units isotropic.

**MM\_ANISOTROPIC:** Logical units are [**anisotropic**](#gt_8356f4cd-ad99-4fb4-a10e-6aeeb1712c26); that is, they are mapped to arbitrary units with arbitrarily scaled axes. The EMR\_SETWINDOWEXTEX and EMR\_SETVIEWPORTEXTEX records are used to specify the units, orientation, and scaling of the axes.

### MetafileVersion Enumeration

The **MetafileVersion** enumeration defines the interoperability version for [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7).

1. typedef enum
2. {
3. META\_FORMAT\_ENHANCED = 0x00010000
4. } MetafileVersion;

**META\_FORMAT\_ENHANCED:** Specifies EMF metafile interoperability.

### MidLine Enumeration

The **MidLine** enumeration defines values for one of the characteristics in the [**PANOSE**](#gt_8290d13f-8670-4b8e-ae6e-1ebb49319792) system for classifying [**typefaces**](#gt_2fff9b0b-4bc1-4d05-81ef-8dde59fd9789).

1. typedef enum
2. {
3. PAN\_ANY = 0x00,
4. PAN\_NO\_FIT = 0x01,
5. PAN\_MIDLINE\_STANDARD\_TRIMMED = 0x02,
6. PAN\_MIDLINE\_STANDARD\_POINTED = 0x03,
7. PAN\_MIDLINE\_STANDARD\_SERIFED = 0x04,
8. PAN\_MIDLINE\_HIGH\_TRIMMED = 0x05,
9. PAN\_MIDLINE\_HIGH\_POINTED = 0x06,
10. PAN\_MIDLINE\_HIGH\_SERIFED = 0x07,
11. PAN\_MIDLINE\_CONSTANT\_TRIMMED = 0x08,
12. PAN\_MIDLINE\_CONSTANT\_POINTED = 0x09,
13. PAN\_MIDLINE\_CONSTANT\_SERIFED = 0x0A,
14. PAN\_MIDLINE\_LOW\_TRIMMED = 0x0B,
15. PAN\_MIDLINE\_LOW\_POINTED = 0x0C,
16. PAN\_MIDLINE\_LOW\_SERIFED = 0x0D
17. } MidLine;

**PAN\_ANY:** Any.

**PAN\_NO\_FIT:** No fit.

**PAN\_MIDLINE\_STANDARD\_TRIMMED:** Standard/trimmed.

**PAN\_MIDLINE\_STANDARD\_POINTED:** Standard/pointed.

**PAN\_MIDLINE\_STANDARD\_SERIFED:** Standard/serifed.

**PAN\_MIDLINE\_HIGH\_TRIMMED:** High/trimmed.

**PAN\_MIDLINE\_HIGH\_POINTED:** High/pointed.

**PAN\_MIDLINE\_HIGH\_SERIFED:** High/serifed.

**PAN\_MIDLINE\_CONSTANT\_TRIMMED:** Constant/trimmed.

**PAN\_MIDLINE\_CONSTANT\_POINTED:** Constant/pointed.

**PAN\_MIDLINE\_CONSTANT\_SERIFED:** Constant/serifed.

**PAN\_MIDLINE\_LOW\_TRIMMED:** Low/trimmed.

**PAN\_MIDLINE\_LOW\_POINTED:** Low/pointed.

**PAN\_MIDLINE\_LOW\_SERIFED:** Low/serifed.

### ModifyWorldTransformMode Enumeration

The **ModifyWorldTransformMode** enumeration defines modes for changing the [**world-space**](#gt_baadb707-1343-413b-9b36-2496b06e9bd7) to [**page-space**](#gt_f530f123-858a-4af7-b626-adea0aeaa75a) [**transform**](#gt_fdca4a96-8e04-4dc0-acad-1d01e1384a5b) that is currently defined in the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45).

1. typedef enum
2. {
3. MWT\_IDENTITY = 0x01,
4. MWT\_LEFTMULTIPLY = 0x02,
5. MWT\_RIGHTMULTIPLY = 0x03,
6. MWT\_SET = 0x04
7. } ModifyWorldTransformMode;

**MWT\_IDENTITY:** Reset the current transform using the identity matrix. In this mode, the specified transform data is ignored.

**MWT\_LEFTMULTIPLY:** Multiply the current transform. In this mode, the specified transform data is the left multiplicand, and the current transform is the right multiplicand.

**MWT\_RIGHTMULTIPLY:** Multiply the current transform. In this mode, the specified transform data is the right multiplicand, and the current transform is the left multiplicand.

**MWT\_SET:** Setthe current transform to the specified transform data.

The transform data is specified as an XForm object (section [2.2.28](#Section_e84107e9bc2b4a1492345d173adc1b59)).

For more information concerning transforms and [**coordinate spaces**](#gt_84184077-d0d9-41e0-acaf-6596aa066813), see [[MSDN-WRLDPGSPC]](https://go.microsoft.com/fwlink/?LinkId=90169).

### PenStyle Enumeration

The **PenStyle** enumeration defines the attributes of pens that can be used in graphics operations. A pen style is a combination of pen type, line style, [**line cap**](#gt_b0341150-ca76-4424-88fe-607e03848dbf), and [**line join**](#gt_4fd3ebb8-54f8-47e8-aded-399a08ad873e).

1. typedef enum
2. {
3. PS\_COSMETIC = 0x00000000,
4. PS\_ENDCAP\_ROUND = 0x00000000,
5. PS\_JOIN\_ROUND = 0x00000000,
6. PS\_SOLID = 0x00000000,
7. PS\_DASH = 0x00000001,
8. PS\_DOT = 0x00000002,
9. PS\_DASHDOT = 0x00000003,
10. PS\_DASHDOTDOT = 0x00000004,
11. PS\_NULL = 0x00000005,
12. PS\_INSIDEFRAME = 0x00000006,
13. PS\_USERSTYLE = 0x00000007,
14. PS\_ALTERNATE = 0x00000008,
15. PS\_ENDCAP\_SQUARE = 0x00000100,
16. PS\_ENDCAP\_FLAT = 0x00000200,
17. PS\_JOIN\_BEVEL = 0x00001000,
18. PS\_JOIN\_MITER = 0x00002000,
19. PS\_GEOMETRIC = 0x00010000
20. } PenStyle;

**PS\_COSMETIC:** A pen type that specifies a line with a width of one logical unit and a style that is a solid color.

**PS\_ENDCAP\_ROUND:** A line cap that specifies round ends.

**PS\_JOIN\_ROUND:** A line join that specifies round joins.

**PS\_SOLID:** A line style that is a solid color.

**PS\_DASH:** A line style that is dashed.

**PS\_DOT:** A line style that is dotted.

**PS\_DASHDOT:** A line style that consists of alternating dashes and dots.

**PS\_DASHDOTDOT:** A line style that consists of dashes and double dots.

**PS\_NULL:** A line style that is invisible.

**PS\_INSIDEFRAME:** A line style that is a solid color. When this style is specified in a drawing record that takes a bounding rectangle, the dimensions of the figure are shrunk so that it fits entirely in the bounding rectangle, considering the width of the pen.

**PS\_USERSTYLE:** A line style that is defined by a styling array, which specifies the lengths of dashes and gaps in the line.

**PS\_ALTERNATE:** A line style in which every other pixel is set. This style is applicable only to a pen type of **PS\_COSMETIC**.

**PS\_ENDCAP\_SQUARE:** A line cap that specifies square ends.

**PS\_ENDCAP\_FLAT:** A line cap that specifies flat ends.

**PS\_JOIN\_BEVEL:** A line join that specifies beveled joins.

**PS\_JOIN\_MITER:** A line join that specifies mitered joins when the lengths of the joins are within the current [**miter length**](#gt_7d78ab9f-9531-4993-b140-8c2bb6e82220) limit. If the lengths of the joins exceed the miter limit, beveled joins are specified.

The miter length limit is a [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7) state property that is set by the EMR\_SETMITERLIMIT record (section [2.3.11.21](#Section_2637d0d092dd48e8be5e6400b4d1fc5e)).

**PS\_GEOMETRIC:** A pen type that specifies a line with a width that is measured in logical units and a style that can contain any of the attributes of a brush.

### Point Enumeration

The Point enumeration is used to specify how a point is to be used in a drawing call.

1. typedef enum
2. {
3. PT\_CLOSEFIGURE = 0x01,
4. PT\_LINETO = 0x02,
5. PT\_BEZIERTO = 0x04,
6. PT\_MOVETO = 0x06
7. } Point;

**PT\_CLOSEFIGURE:** A **PT\_LINETO** or **PT\_BEZIERTO** type can be combined with this value by using the bitwise operator OR to indicate that the corresponding point is the last point in a figure and the figure is closed.

The current position is set to the ending point of the closing line.

**PT\_LINETO:** Specifies that a line is to be drawn from the current position to this point, which then becomes the new current position.

**PT\_BEZIERTO:** Specifies that this point is a control point or ending point for a [**Bezier curve**](#gt_b12730d5-b22e-4027-ba59-ff53749022a5).

**PT\_BEZIERTO** types always occur in sets of three. The current position defines the starting point for the Bezier curve. The first two **PT\_BEZIERTO** points are the control points, and the third **PT\_BEZIERTO** point is the ending point. The ending point becomes the new current position. If there are not three consecutive **PT\_BEZIERTO** points, an error results.

**PT\_MOVETO:** Specifies that this point starts a disjoint figure. This point becomes the new current position.

### PolygonFillMode Enumeration

The PolygonFillMode enumeration defines values that specify how to calculate the [**region**](#gt_6f13fe5f-8747-4ae0-9375-814bf0528197) of a polygon that is to be filled.

1. typedef enum
2. {
3. ALTERNATE = 0x01,
4. WINDING = 0x02
5. } PolygonFillMode;

**ALTERNATE:** Selects alternate mode (fills the area between odd-numbered and even-numbered polygon sides on each scan line).

**WINDING:** Selects winding mode (fills any region with a nonzero winding value).

### Proportion Enumeration

The Proportion enumeration defines values for one of the characteristics in the [**PANOSE**](#gt_8290d13f-8670-4b8e-ae6e-1ebb49319792) system for classifying [**typefaces**](#gt_2fff9b0b-4bc1-4d05-81ef-8dde59fd9789).

1. typedef enum
2. {
3. PAN\_ANY = 0x00,
4. PAN\_NO\_FIT = 0x01,
5. PAN\_PROP\_OLD\_STYLE = 0x02,
6. PAN\_PROP\_MODERN = 0x03,
7. PAN\_PROP\_EVEN\_WIDTH = 0x04,
8. PAN\_PROP\_EXPANDED = 0x05,
9. PAN\_PROP\_CONDENSED = 0x06,
10. PAN\_PROP\_VERY\_EXPANDED = 0x07,
11. PAN\_PROP\_VERY\_CONDENSED = 0x08,
12. PAN\_PROP\_MONOSPACED = 0x09
13. } Proportion;

**PAN\_ANY:** Any.

**PAN\_NO\_FIT:** No fit.

**PAN\_PROP\_OLD\_STYLE:** Old style.

**PAN\_PROP\_MODERN:** Modern.

**PAN\_PROP\_EVEN\_WIDTH:** Even width.

**PAN\_PROP\_EXPANDED:** Expanded.

**PAN\_PROP\_CONDENSED:** Condensed.

**PAN\_PROP\_VERY\_EXPANDED:** Very expanded.

**PAN\_PROP\_VERY\_CONDENSED:** Very condensed.

**PAN\_PROP\_MONOSPACED:** Monospaced.

### RegionMode Enumeration

The **RegionMode** enumeration defines values that are used with [EMR\_SELECTCLIPPATH](#Section_4a26bcf866074a098ec3a8768eadc8e8) and [EMR\_EXTSELECTCLIPRGN](#Section_c6b9f4e627f64a4da383c2daf5da11d9), specifying the current [**path bracket**](#gt_b89ea7da-c0bf-4c45-9f3e-fc430d7ab601) or a new [**region**](#gt_6f13fe5f-8747-4ae0-9375-814bf0528197) that is being combined with the current clipping region.

1. typedef enum
2. {
3. RGN\_AND = 0x01,
4. RGN\_OR = 0x02,
5. RGN\_XOR = 0x03,
6. RGN\_DIFF = 0x04,
7. RGN\_COPY = 0x05
8. } RegionMode;

**RGN\_AND:** The new clipping region includes the intersection (overlapping areas) of the current clipping region and the current path bracket (or new region).

**RGN\_OR:** The new clipping region includes the union (combined areas) of the current clipping region and the current path bracket (or new region).

**RGN\_XOR:** The new clipping region includes the union of the current clipping region and the current path bracket (or new region) but without the overlapping areas.

**RGN\_DIFF:** The new clipping region includes the areas of the current clipping region with those of the current path bracket (or new region) excluded.

**RGN\_COPY:** The new clipping region is the current path bracket (or the new region).

### SerifType Enumeration

The SerifType enumeration defines values for one of the characteristics in the [**PANOSE**](#gt_8290d13f-8670-4b8e-ae6e-1ebb49319792) system for classifying [**typefaces**](#gt_2fff9b0b-4bc1-4d05-81ef-8dde59fd9789).

1. typedef enum
2. {
3. PAN\_ANY = 0x00,
4. PAN\_NO\_FIT = 0x01,
5. PAN\_SERIF\_COVE = 0x02,
6. PAN\_SERIF\_OBTUSE\_COVE = 0x03,
7. PAN\_SERIF\_SQUARE\_COVE = 0x04,
8. PAN\_SERIF\_OBTUSE\_SQUARE\_COVE = 0x05,
9. PAN\_SERIF\_SQUARE = 0x06,
10. PAN\_SERIF\_THIN = 0x07,
11. PAN\_SERIF\_BONE = 0x08,
12. PAN\_SERIF\_EXAGGERATED = 0x09,
13. PAN\_SERIF\_TRIANGLE = 0x0A,
14. PAN\_SERIF\_NORMAL\_SANS = 0x0B,
15. PAN\_SERIF\_OBTUSE\_SANS = 0x0C,
16. PAN\_SERIF\_PERP\_SANS = 0x0D,
17. PAN\_SERIF\_FLARED = 0x0E,
18. PAN\_SERIF\_ROUNDED = 0x0F
19. } SerifType;

**PAN\_ANY:** Any.

**PAN\_NO\_FIT:** No fit.

**PAN\_SERIF\_COVE:** Cove.

**PAN\_SERIF\_OBTUSE\_COVE:** Obtuse cove.

**PAN\_SERIF\_SQUARE\_COVE:** Square cove.

**PAN\_SERIF\_OBTUSE\_SQUARE\_COVE:** Obtuse square cove.

**PAN\_SERIF\_SQUARE:** Square.

**PAN\_SERIF\_THIN:** Thin.

**PAN\_SERIF\_BONE:** Bone.

**PAN\_SERIF\_EXAGGERATED:** Exaggerated.

**PAN\_SERIF\_TRIANGLE:** Triangle.

**PAN\_SERIF\_NORMAL\_SANS:** Normal sans.

**PAN\_SERIF\_OBTUSE\_SANS:** Obtuse sans.

**PAN\_SERIF\_PERP\_SANS:** Perp sans.

**PAN\_SERIF\_FLARED:** Flared.

**PAN\_SERIF\_ROUNDED:** Rounded.

### StockObject Enumeration

The **StockObject** enumeration specifies the indexes of predefined logical graphics objects that can be used in graphics operations.

The specific structures of [**stock objects**](#gt_b1328cb9-acd8-47f8-9896-fe985af2d6d6) are implementation-dependent; however, the properties of stock objects SHOULD be equivalent to the properties of explicitly created objects of the same type. These properties are specified where possible for the stock objects defined in this enumeration.

1. typedef enum
2. {
3. WHITE\_BRUSH = 0x80000000,
4. LTGRAY\_BRUSH = 0x80000001,
5. GRAY\_BRUSH = 0x80000002,
6. DKGRAY\_BRUSH = 0x80000003,
7. BLACK\_BRUSH = 0x80000004,
8. NULL\_BRUSH = 0x80000005,
9. WHITE\_PEN = 0x80000006,
10. BLACK\_PEN = 0x80000007,
11. NULL\_PEN = 0x80000008,
12. OEM\_FIXED\_FONT = 0x8000000A,
13. ANSI\_FIXED\_FONT = 0x8000000B,
14. ANSI\_VAR\_FONT = 0x8000000C,
15. SYSTEM\_FONT = 0x8000000D,
16. DEVICE\_DEFAULT\_FONT = 0x8000000E,
17. DEFAULT\_PALETTE = 0x8000000F,
18. SYSTEM\_FIXED\_FONT = 0x80000010,
19. DEFAULT\_GUI\_FONT = 0x80000011,
20. DC\_BRUSH = 0x80000012,
21. DC\_PEN = 0x80000013
22. } StockObject;

**WHITE\_BRUSH:** A white, solid-color brush that is equivalent to a logical brush (LogBrushEx object, section [2.2.12](#Section_6d790f8d51654746bac33443915c2071)) with the following properties:

* **BrushStyle**: BS\_SOLID from the BrushStyle enumeration ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.1.1.4)
* **Color**: 0x00FFFFFF in a ColorRef object ([MS-WMF] section 2.2.2.8)

**LTGRAY\_BRUSH:** A light gray, solid-color brush that is equivalent to a logical brush with the following properties:

* **BrushStyle**: BS\_SOLID
* **Color**: 0x00C0C0C0

**GRAY\_BRUSH:** A gray, solid-color brush that is equivalent to a logical brush with the following properties:

* **BrushStyle**: BS\_SOLID
* **Color**: 0x00808080

**DKGRAY\_BRUSH:** A dark gray, solid color brush that is equivalent to a logical brush with the following properties:

* **BrushStyle**: BS\_SOLID
* **Color**: 0x00404040

**BLACK\_BRUSH:** A black, solid color brush that is equivalent to a logical brush with the following properties:

* **BrushStyle**: BS\_SOLID
* **Color**: 0x00000000

**NULL\_BRUSH:** A null brush that is equivalent to a logical brush with the following properties:

* **BrushStyle**: BS\_NULL

**WHITE\_PEN:** A white, solid-color pen that is equivalent to a logical pen (LogPen object, section [2.2.19](#Section_93ce3f4537ac4affb6e82f6db054c4c4)) with the following properties:

* **PenStyle**: PS\_COSMETIC + PS\_SOLID from the PenStyle enumeration (section [2.1.25](#Section_b2e629158401429fbb6db2506a4c2fe8))
* **ColorRef**: 0x00FFFFFF in a ColorRef object.

**BLACK\_PEN:** A black, solid-color pen that is equivalent to a logical pen with the following properties:

* **PenStyle**: PS\_COSMETIC + PS\_SOLID
* **ColorRef**: 0x00000000

**NULL\_PEN:** A null pen that is equivalent to a logical pen with the following properties:

* **PenStyle**: PS\_NULL

**OEM\_FIXED\_FONT:** A fixed-width, [**OEM character set**](#gt_3240e34e-920e-40ac-a672-342ac34a5e22) font that is equivalent to a LogFont object (section [2.2.13](#Section_1cd7cb7a1a7c48d2b4a4c218f9d12100)) with the following properties:

* **Charset**: OEM\_CHARSET from the CharacterSet enumeration ([MS-WMF] section 2.1.1.5)
* **PitchAndFamily**: **FF\_DONTCARE** (FamilyFont enumeration, [MS-WMF] section 2.1.1.8) + **FIXED\_PITCH** (PitchFont enumeration, [MS-WMF] section 2.1.1.24)

**ANSI\_FIXED\_FONT:** A fixed-width font that is equivalent to a LogFont object with the following properties:[<33>](#Appendix_A_33" \o "Product behavior note 33)

* **Charset**: ANSI\_CHARSET
* **PitchAndFamily**: FF\_DONTCARE + FIXED\_PITCH

**ANSI\_VAR\_FONT:** A variable-width font that is equivalent to a logical font with the following properties:[<34>](#Appendix_A_34" \o "Product behavior note 34)

* **Charset**: ANSI\_CHARSET
* **PitchAndFamily**: FF\_DONTCARE + VARIABLE\_PITCH

**SYSTEM\_FONT:** A font that is guaranteed to be available in the operating system. The actual font that is specified by this value is implementation-dependent.[<35>](#Appendix_A_35" \o "Product behavior note 35)

**DEVICE\_DEFAULT\_FONT:** The default font that is provided by the graphics device driver for the current output device. The actual font that is specified by this value is implementation-dependent.[<36>](#Appendix_A_36" \o "Product behavior note 36)

**DEFAULT\_PALETTE:** The default [**palette**](#gt_dca5cd91-8d01-424b-9722-c9e818562ee5) that is defined for the current output device. The actual palette that is specified by this value is implementation-dependent.[<37>](#Appendix_A_37" \o "Product behavior note 37)

**SYSTEM\_FIXED\_FONT:** A fixed-width font that is guaranteed to be available in the operating system. The actual font that is specified by this value is implementation-dependent.

**DEFAULT\_GUI\_FONT:** The default font that is used for user interface objects such as menus and dialog boxes. The actual font that is specified by this value is implementation-dependent.[<38>](#Appendix_A_38" \o "Product behavior note 38)

**DC\_BRUSH:** The solid-color brush that is currently selected in the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45). The default SHOULD[<39>](#Appendix_A_39" \o "Product behavior note 39) be WHITE\_BRUSH.

**DC\_PEN:** The solid-color pen that is currently selected in the playback device context. The default SHOULD[<40>](#Appendix_A_40" \o "Product behavior note 40) be BLACK\_PEN.

During [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7) processing, stock object indexes can be used by object manipulation records (section [2.3.8](#Section_2f1c3268b43b4c28988b41aa49354998)) in the same way as indexes of graphics objects that are explicitly created by object creation records (section [2.3.7](#Section_7d914464d5464393bddb6d1e747fca18)). The index of a stock object can be distinguished from the index of an explicit object by the value of the most-significant bit. If that bit is set, the object is a stock object; if the bit is clear, the object was created by a previous metafile record.

### StretchMode Enumeration

The **StretchMode** enumeration is used to specify how color data is added to or removed from [**bitmaps**](#gt_26a17e58-eeba-4210-81b9-8bbadf7ee952) that are stretched or compressed.[<41>](#Appendix_A_41" \o "Product behavior note 41)

1. typedef enum
2. {
3. STRETCH\_ANDSCANS = 0x01,
4. STRETCH\_ORSCANS = 0x02,
5. STRETCH\_DELETESCANS = 0x03,
6. STRETCH\_HALFTONE = 0x04
7. } StretchMode;

**STRETCH\_ANDSCANS:** Performs a [**Boolean**](#gt_1d79d7a7-ba2c-4b34-931c-7ba8057c87b2) AND operation using the color values for the eliminated and existing pixels. If the bitmap is a monochrome bitmap, this mode preserves black pixels at the expense of white pixels.

**STRETCH\_ORSCANS:** Performs a Boolean OR operation using the color values for the eliminated and existing pixels. If the bitmap is a monochrome bitmap, this mode preserves white pixels at the expense of black pixels.

**STRETCH\_DELETESCANS:** Deletes the pixels. This mode deletes all eliminated lines of pixels without trying to preserve their information.

**STRETCH\_HALFTONE:** Maps pixels from the source rectangle into blocks of pixels in the destination rectangle. The average color over the destination block of pixels approximates the color of the source pixels.

After setting the **STRETCH\_HALFTONE** stretching mode, the brush origin SHOULD be defined by an [EMR\_SETBRUSHORGEX](#Section_b02ab7ee4c884bf29ce99100636d86be) record. If it fails to do so, brush misalignment can occur.

### StrokeVariation Enumeration

The StrokeVariation enumeration defines values for one of the characteristics in the [**PANOSE**](#gt_8290d13f-8670-4b8e-ae6e-1ebb49319792) system for classifying [**typefaces**](#gt_2fff9b0b-4bc1-4d05-81ef-8dde59fd9789).

1. typedef enum
2. {
3. PAN\_ANY = 0x00,
4. PAN\_NO\_FIT = 0x01,
5. PAN\_STROKE\_GRADUAL\_DIAG = 0x02,
6. PAN\_STROKE\_GRADUAL\_TRAN = 0x03,
7. PAN\_STROKE\_GRADUAL\_VERT = 0x04,
8. PAN\_STROKE\_GRADUAL\_HORZ = 0x05,
9. PAN\_STROKE\_RAPID\_VERT = 0x06,
10. PAN\_STROKE\_RAPID\_HORZ = 0x07,
11. PAN\_STROKE\_INSTANT\_VERT = 0x08
12. } StrokeVariation;

**PAN\_ANY:** Any.

**PAN\_NO\_FIT:** No fit.

**PAN\_STROKE\_GRADUAL\_DIAG:** Gradual/diagonal.

**PAN\_STROKE\_GRADUAL\_TRAN:** Gradual/transitional.

**PAN\_STROKE\_GRADUAL\_VERT:** Gradual/vertical.

**PAN\_STROKE\_GRADUAL\_HORZ:** Gradual/horizontal.

**PAN\_STROKE\_RAPID\_VERT:** Rapid/vertical.

**PAN\_STROKE\_RAPID\_HORZ:** Rapid/horizontal.

**PAN\_STROKE\_INSTANT\_VERT:** Instant/vertical.

### Weight Enumeration

The Weight enumeration defines values for one of the characteristics in the [**PANOSE**](#gt_8290d13f-8670-4b8e-ae6e-1ebb49319792) system for classifying [**typefaces**](#gt_2fff9b0b-4bc1-4d05-81ef-8dde59fd9789).

1. typedef enum
2. {
3. PAN\_ANY = 0x00,
4. PAN\_NO\_FIT = 0x01,
5. PAN\_WEIGHT\_VERY\_LIGHT = 0x02,
6. PAN\_WEIGHT\_LIGHT = 0x03,
7. PAN\_WEIGHT\_THIN = 0x04,
8. PAN\_WEIGHT\_BOOK = 0x05,
9. PAN\_WEIGHT\_MEDIUM = 0x06,
10. PAN\_WEIGHT\_DEMI = 0x07,
11. PAN\_WEIGHT\_BOLD = 0x08,
12. PAN\_WEIGHT\_HEAVY = 0x09,
13. PAN\_WEIGHT\_BLACK = 0x0A,
14. PAN\_WEIGHT\_NORD = 0x0B
15. } Weight;

**PAN\_ANY:** Any.

**PAN\_NO\_FIT:** No fit.

**PAN\_WEIGHT\_VERY\_LIGHT:** Very light.

**PAN\_WEIGHT\_LIGHT:** Light.

**PAN\_WEIGHT\_THIN:** Thin.

**PAN\_WEIGHT\_BOOK:** Book.

**PAN\_WEIGHT\_MEDIUM:** Medium.

**PAN\_WEIGHT\_DEMI:** Demi.

**PAN\_WEIGHT\_BOLD:** Bold.

**PAN\_WEIGHT\_HEAVY:** Heavy.

**PAN\_WEIGHT\_BLACK:** Black.

**PAN\_WEIGHT\_NORD:** Nord.

### XHeight Enumeration

The XHeight enumeration defines values for one of the characteristics in the [**PANOSE**](#gt_8290d13f-8670-4b8e-ae6e-1ebb49319792) system for classifying [**typefaces**](#gt_2fff9b0b-4bc1-4d05-81ef-8dde59fd9789).

1. typedef enum
2. {
3. PAN\_ANY = 0x00,
4. PAN\_NO\_FIT = 0x01,
5. PAN\_XHEIGHT\_CONSTANT\_SMALL = 0x02,
6. PAN\_XHEIGHT\_CONSTANT\_STD = 0x03,
7. PAN\_XHEIGHT\_CONSTANT\_LARGE = 0x04,
8. PAN\_XHEIGHT\_DUCKING\_SMALL = 0x05,
9. PAN\_XHEIGHT\_DUCKING\_STD = 0x06,
10. PAN\_XHEIGHT\_DUCKING\_LARGE = 0x07
11. } XHeight;

**PAN\_ANY:** Any.

**PAN\_NO\_FIT:** No fit.

**PAN\_XHEIGHT\_CONSTANT\_SMALL:** Constant/small.

**PAN\_XHEIGHT\_CONSTANT\_STD:** Constant/standard.

**PAN\_XHEIGHT\_CONSTANT\_LARGE:** Constant/large.

**PAN\_XHEIGHT\_DUCKING\_SMALL:** [**Ducking**](#gt_85a2a285-eeff-4059-82e9-d924141f244e)/small

**PAN\_XHEIGHT\_DUCKING\_STD:** Ducking/standard.

**PAN\_XHEIGHT\_DUCKING\_LARGE:** Ducking/large.

## EMF Objects

### BitFIX28\_4 Object

The BitFIX28\_4 object defines a numeric value in [**28.4 bit FIX notation**](#gt_1bcc53a6-1861-49e9-bd00-3a38a888d9f4).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| IntValue | | | | | | | | | | | | | | | | | | | | | | | | | | | | FracValue | | | |

**IntValue (28 bits):** The signed, integral part of the number.

**FracValue (4 bits):** The unsigned fractional part of the number, in units of one-sixteenth.

The real number represented by this object is computed as follows:

1. IntValue + (FracValue / 16)

### ColorAdjustment Object

The ColorAdjustment object defines values for adjusting the colors in source [**bitmaps**](#gt_26a17e58-eeba-4210-81b9-8bbadf7ee952) in bit-block transfers.[<42>](#Appendix_A_42" \o "Product behavior note 42)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Size | | | | | | | | | | | | | | | | Values | | | | | | | | | | | | | | | |
| IlluminantIndex | | | | | | | | | | | | | | | | RedGamma | | | | | | | | | | | | | | | |
| GreenGamma | | | | | | | | | | | | | | | | BlueGamma | | | | | | | | | | | | | | | |
| ReferenceBlack | | | | | | | | | | | | | | | | ReferenceWhite | | | | | | | | | | | | | | | |
| Contrast | | | | | | | | | | | | | | | | Brightness | | | | | | | | | | | | | | | |
| Colorfulness | | | | | | | | | | | | | | | | RedGreenTint | | | | | | | | | | | | | | | |

**Size (2 bytes):** An unsigned integer that specifies the size in bytes of this object. This value is 0x0018.

**Values (2 bytes):** An unsigned integer that specifies how to prepare the output image. This field can be set to NULL or to any combination of values in the [ColorAdjustment](#Section_c80a94d5415d4c4c83b812c0eb98b81f) enumeration (section 2.1.5).

**IlluminantIndex (2 bytes):** An unsigned integer that specifies the type of standard light source under which the image is viewed, from the [Illuminant](#Section_7cd7679e4a1445d19cacbf7f80be2c10) enumeration (section 2.1.19).

**RedGamma (2 bytes):** An unsigned integer that specifies the nth power [**gamma correction**](#gt_60eaec33-bd4f-4e71-93e6-4ef382aa80f9) value for the red primary of the source colors. This value SHOULD be in the range from 2,500 to 65,000.[<43>](#Appendix_A_43" \o "Product behavior note 43) A value of 10,000 means gamma correction MUST NOT be performed.

**GreenGamma (2 bytes):** An unsigned integer that specifies the nth power gamma correction value for the green primary of the source colors. This value SHOULD be in the range from 2,500 to 65,000. A value of 10,000 means gamma correction MUST NOT be performed.

**BlueGamma (2 bytes):** An unsigned integer that specifies the nth power gamma correction value for the blue primary of the source colors. This value SHOULD be in the range from 2,500 to 65,000. A value of 10,000 means gamma correction MUST NOT be performed.

**ReferenceBlack (2 bytes):** An unsigned integer that specifies the black reference for the source colors. Any colors that are darker than this are treated as black. This value SHOULD be in the range from zero to 4,000.

**ReferenceWhite (2 bytes):** An unsigned integer that specifies the white reference for the source colors. Any colors that are lighter than this are treated as white. This value SHOULD be in the range from 6,000 to 10,000.

**Contrast (2 bytes):** A signed integer that specifies the amount of [**contrast**](#gt_ceb96b0c-3f4c-4ad8-aa79-289d935ac577) to be applied to the source object. This value SHOULD be in the range from –100 to 100. A value of zero means contrast adjustment MUST NOT be performed.

**Brightness (2 bytes):** A signed integer that specifies the amount of [**brightness**](#gt_227798e1-af7a-4ab5-b3a9-c73f02d9c504) to be applied to the source object. This value SHOULD be in the range from –100 to 100. A value of zero means brightness adjustment MUST NOT be performed.

**Colorfulness (2 bytes):** A signed integer that specifies the amount of [**colorfulness**](#gt_0140633b-37c2-4260-bae2-ca578de1b278) to be applied to the source object. This value SHOULD be in the range from –100 to 100. A value of zero means colorfulness adjustment MUST NOT be performed.

**RedGreenTint (2 bytes):** A signed integer that specifies the amount of red or green [**tint**](#gt_ff6845a6-3de9-4eaf-a14d-0b3137c58691) adjustment to be applied to the source object. This value SHOULD be in the range from –100 to 100. Positive numbers adjust towards red and negative numbers adjust towards green. A value of zero means tint adjustment MUST NOT be performed.

The ColorAdjustment object is used in bit-block transfers performed by [EMR\_STRETCHBLT](#Section_0dd551f380f7485289c29ddba803a192) and [EMR\_STRETCHDIBITS](#Section_89c0d8080dea413fbe402e9e51fa36ac) records when the [StretchMode](#Section_e72f20aa2e2d453f9e95eaef50b76e19) enumeration (section 2.1.32) value is **STRETCH\_HALFTONE**. The color adjustment values can apply a color filter or lighten or darken an image.

An [EMR\_SETCOLORADJUSTMENT](#Section_81b3cd7ced6742aaab9847b99d697231) record (section 2.3.11.13) sets the current ColorAdjustment object in the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45). That ColorAdjustment object affects all subsequent EMR\_STRETCHBLT and EMR\_STRETCHDIBITS records until a different ColorAdjustment object is specified by another EMR\_SETCOLORADJUSTMENT record, or until the object is removed by a [EMR\_DELETEOBJECT](#Section_6f0f12a3111a478b8251a9505168f9a9) record.

### DesignVector Object

The DesignVector (section 2.2.3) object defines the [**design vector**](#gt_68697457-d046-45a9-b822-86cb2b8c3320), which specifies values for the [**font axes**](#gt_5d727201-0a47-44b1-86e5-faf77f0f2463) of a [**multiple master**](#gt_fef5ee7a-ad0c-4395-a1ac-f130405eec03) font.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Signature | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NumAxes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Values (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Signature (4 bytes):** An unsigned integer that MUST be set to the value 0x08007664.

**NumAxes (4 bytes):** An unsigned integer that specifies the number of elements in the **Values** array. It MUST be in the range 0 to 16, inclusive.

**Values (variable, optional):** An array of 32-bit signed integers that specify the values of the font axes of a multiple master, [**OpenType**](#gt_5e824598-05ae-44e2-99bd-78c347279ad9) font. The maximum number of values in the array is 16.

### EmrFormat Object

The EmrFormat object contains information that identifies the format of image data in an [EMR\_COMMENT\_MULTIFORMATS record (section 2.3.3.4.3)](#Section_4bc4608eca82426dac0e5ded2ad3fc4e).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Signature | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Version | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SizeData | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offData | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Signature (4 bytes):** An unsigned integer that specifies the format of the image data. This value is in the [FormatSignature enumeration (section 2.1.14)](#Section_b55cbd24612d47fe81c91da6a682246f).

**Version (4 bytes):** An unsigned integer that specifies the format version number. If the **Signature** field specifies [**encapsulated PostScript (EPS)**](#gt_ba716063-a4a2-4e6e-a657-7ba9fde7b93f), this value is 0x00000001; otherwise, this value is ignored.

**SizeData (4 bytes):** An unsigned integer that specifies the size of the data in bytes.

**offData (4 bytes):** An unsigned integer that specifies the offset to the data from the start of the **identifier** field in an [EMR\_COMMENT\_PUBLIC record (section 2.3.3.4)](#Section_39c91f22e4cc4e82a6050137caa8457c). The offset MUST be 32-bit aligned.

### EmrText Object

The **EmrText** object contains values for text output.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Reference | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Chars | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offString | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Options | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Rectangle (optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offDx | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| StringBuffer (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DxBuffer (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Reference (8 bytes):** A PointL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.15) that specifies the coordinates of the reference point used to position the string. The reference point is defined by the last EMR\_SETTEXTALIGN record (section [2.3.11.25](#Section_a49bb565a58341da8d4a8a0314b3e397)). If no such record has been set, the default alignment is (TA\_LEFT, TA\_TOP), which is specified using TextAlignmentMode flags ([MS-WMF] section 2.1.2.3).

**Chars (4 bytes):** An unsigned integer that specifies the number of characters in the string.

**offString (4 bytes):** An unsigned integer that specifies the offset to the output string in bytes, from the start of the record in which this object is contained. This value is 8- or 16-bit aligned, according to the character format.

**Options (4 bytes):** An unsigned integer that specifies how to use the rectangle specified in the **Rectangle** field. This field can be a combination of more than one ExtTextOutOptions enumeration (section [2.1.11](#Section_e7ffcc5340d148738edac5c5ee104aa5)) values.

**Rectangle (16 bytes, optional):** A RectL object ([MS-WMF] section 2.2.2.19) that defines a clipping and/or opaquing rectangle in logical units. This rectangle is applied to the text output performed by the containing record.[<44>](#Appendix_A_44" \o "Product behavior note 44)

**offDx (4 bytes):** An unsigned integer that specifies the offset to an intercharacter spacing array in bytes, from the start of the record in which this object is contained. This value is 32-bit aligned.

**StringBuffer (variable):** The character string buffer.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| UndefinedSpace1 (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OutputString (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**UndefinedSpace1 (variable, optional):** The number of unused bytes. The **OutputString** field is not required to follow immediately the preceding portion of this structure.

**OutputString (variable):** An array of characters that specify the string to output. The location of this field is specified by the value of **offString** in bytes from the start of this record. The number of characters is specified by the value of **Chars**.

**DxBuffer (variable, optional):** The character spacing buffer.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| UndefinedSpace2 (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OutputDx (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**UndefinedSpace2 (variable, optional):** The number of unused bytes. The **OutputDx** field is not required to follow immediately the preceding portion of this structure.

**OutputDx (variable):** An array of 32-bit unsigned integers that specify the output spacing between the origins of adjacent character cells in logical units. The location of this field is specified by the value of **offDx** in bytes from the start of this record. If spacing is defined, this field contains the same number of values as characters in the output string.

If the Options field of the EmrText object contains the ETO\_PDY flag, then this buffer contains twice as many values as there are characters in the output string, one horizontal and one vertical offset for each, in that order.

If ETO\_RTLREADING is specified, characters are laid right to left instead of left to right. No other options affect the interpretation of this field.

The size and encoding of the characters in the **OutputString** is determined by the type of record that contains this object, as follows:

* EMR\_EXTTEXTOUTA (section [2.3.5.7](#Section_6b582a713c294fc6a0f41f8a313739a1)) and EMR\_POLYTEXTOUTA (section [2.3.5.32](#Section_d8b7ac2176b84f9aa7cc05f9d2d5627e)) records: 8-bit [**ASCII**](#gt_79fa85ca-ac61-467c-b819-e97dc1a7a599) characters.
* EMR\_EXTTEXTOUTW (section [2.3.5.8](#Section_a59a79ac328e492da34de02727af6edf)) and EMR\_POLYTEXTOUTW (section [2.3.5.33](#Section_381015eb29a7477493ada210697f5972)) records: 16-bit [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) [**UTF16-LE**](#gt_f25550c9-f84f-4eb2-8156-14794a7e3059) characters.

### EpsData Object

The EpsData object is a container for [**EPS**](#gt_ba716063-a4a2-4e6e-a657-7ba9fde7b93f) data.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| SizeData | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Version | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Points (24 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PostScriptData (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**SizeData (4 bytes):** An unsigned integer that specifies the total size of this object in bytes.

**Version (4 bytes):** An unsigned integer that specifies the [**PostScript**](#gt_31971c63-f974-47de-8de9-6b76d093890d) language level. This value is 0x00000001.

**Points (24 bytes):** An array of three [Point28\_4 objects (section 2.2.23)](#Section_c66bda3b65ae4b3f8c8f62ad7f94f730) that defines the coordinates of the output parallelogram using [**28.4 bit FIX notation**](#gt_1bcc53a6-1861-49e9-bd00-3a38a888d9f4).

The upper-left corner of the parallelogram is the first point in this array, the upper-right corner is the second point, and the lower-left corner is the third point. The lower-right corner of the parallelogram is computed from the first three points (A, B, and C) by treating them as vectors.

1. D = B + C A

**PostScriptData (variable):** An array of bytes of PostScript data. The length of this array can be computed from the **SizeData** field. This data MAY be used to render an image.[<45>](#Appendix_A_45" \o "Product behavior note 45)

An EpsData object can be used to embed a PostScript image in an [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7) as follows:

* An EMF metafile contains an [EMR\_COMMENT\_MULTIFORMATS record (section 2.3.3.4.3)](#Section_4bc4608eca82426dac0e5ded2ad3fc4e).
* The EMR\_COMMENT\_MULTIFORMATS record specifies an **aFormats** field that contains an [EmrFormat object (section 2.2.4)](#Section_1bd63563ce674528bf285a8852ff0270).
* The EmrFormat object specifies a **Signature** field that is set to **EPS\_SIGNATURE** from the [FormatSignature enumeration (section 2.1.14)](#Section_b55cbd24612d47fe81c91da6a682246f).
* The **EPS\_SIGNATURE** value specifies that the **FormatData** field in the EMR\_COMMENT\_MULTIFORMATS record contains an EpsData object.
* The EmrFormat object also specifies an **offData** field that indicates where the EpsData object is in the **FormatData** field in the EMR\_COMMENT\_MULTIFORMATS record.

### GradientRectangle Object

The GradientRectangle object defines a rectangle using [TriVertex objects (section 2.2.26)](#Section_33578509834946b68f8f107c3f70bace) in an [EMR\_GRADIENTFILL record (section 2.3.5.12)](#Section_1a3849c8be6c4d30b5d3f43b4c70ca0d).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| UpperLeft | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| LowerRight | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**UpperLeft (4 bytes):** An index into an array of TriVertex objects that specifies the upper-left vertex of a rectangle. The index MUST be smaller than the size of the array, as defined by the **nVer** field of the EMR\_GRADIENTFILL record.

**LowerRight (4 bytes):** An index into an array of TriVertex objects that specifies the lower-right vertex of a rectangle. The index MUST be smaller than the size of the array, as defined by the **nVer** field of the EMR\_GRADIENTFILL record.

### GradientTriangle Object

The GradientTriangle object defines a triangle using [TriVertex objects (section 2.2.26)](#Section_33578509834946b68f8f107c3f70bace) in an [EMR\_GRADIENTFILL record (section 2.3.5.12)](#Section_1a3849c8be6c4d30b5d3f43b4c70ca0d).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Vertex1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Vertex2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Vertex3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Vertex1 (4 bytes):** An index into an array of TriVertex objects that specifies a vertex of a triangle. The index MUST be smaller than the size of the array, as defined by the **nVer** field of the EMR\_GRADIENTFILL record.

**Vertex2 (4 bytes):** An index into an array of TriVertex objects that specifies a vertex of a triangle. The index MUST be smaller than the size of the array, as defined by the **nVer** field of the EMR\_GRADIENTFILL record.

**Vertex3 (4 bytes):** An index into an array of TriVertex objects that specifies a vertex of a triangle. The index MUST be smaller than the size of the array, as defined by the **nVer** field of the EMR\_GRADIENTFILL record.

### Header Object

The Header object defines the [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7) header. It specifies properties of the device on which the image in the metafile was created.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Frame | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| RecordSignature | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Version | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Records | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Handles | | | | | | | | | | | | | | | | Reserved | | | | | | | | | | | | | | | |
| nDescription | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offDescription | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| nPalEntries | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Device | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Millimeters | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Bounds (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19) that specifies the rectangular [**inclusive-inclusive**](#gt_c706d1bc-8878-4743-9b94-e9643c5269e4) bounds in logical units of the smallest rectangle that can be drawn around the image stored in the metafile.

**Frame (16 bytes):** A RectL object that specifies the rectangular inclusive-inclusive dimensions, in .01 millimeter units, of a rectangle that surrounds the image stored in the metafile.

**RecordSignature (4 bytes):** An unsigned integer that specifies the record signature. This MUST be **ENHMETA\_SIGNATURE**, from the FormatSignature enumeration (section [2.1.14](#Section_b55cbd24612d47fe81c91da6a682246f)).

**Version (4 bytes):** An unsigned integer that specifies the EMF version for interoperability. This MAY be 0x00010000.

**Bytes (4 bytes):** An unsigned integer that specifies the size of the metafile in bytes.

**Records (4 bytes):** An unsigned integer that specifies the number of records in the metafile.

**Handles (2 bytes):** An unsigned integer that specifies the number of graphics objects that are used during the processing of the metafile.

**Reserved (2 bytes):** An unsigned integer that MUST be 0x0000 and MUST be ignored.

**nDescription (4 bytes):** An unsigned integer that specifies the number of characters in the array that contains the description of the metafile's contents. This is zero if there is no description string.

**offDescription (4 bytes):** An unsigned integer that specifies the offset from the beginning of this record to the array that contains the description of the metafile's contents.

**nPalEntries (4 bytes):** An unsigned integer that specifies the number of entries in the metafile [**palette**](#gt_dca5cd91-8d01-424b-9722-c9e818562ee5). The palette is located in the [EMR\_EOF](#Section_3f47fde00e6b40c187f3f4129af03aa1) record.

**Device (8 bytes):** A SizeL object ([MS-WMF] section 2.2.2.22) that specifies the size of the reference device, in pixels.

**Millimeters (8 bytes):** A SizeL object that specifies the size of the reference device, in millimeters.

### HeaderExtension1 Object

The HeaderExtension1 object defines the first extension to the [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7) header. It adds support for a [PixelFormatDescriptor](#Section_1db036d62da84b92b4f8e9cab8cc93b7) object (section 2.2.22) and [**OpenGL**](#gt_216becb9-a800-4a44-8206-2ec2231c9c67) [[OPENGL]](https://go.microsoft.com/fwlink/?LinkId=90241) records (section [2.3.9](#Section_4a6c2fc239af43b48254449f5f1c5c05)).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| cbPixelFormat | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offPixelFormat | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| bOpenGL | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**cbPixelFormat (4 bytes):** An unsigned integer that specifies the size of the PixelFormatDescriptor object. This value is 0x00000000 if no pixel format is set.

**offPixelFormat (4 bytes):** An unsigned integer that specifies the offset to the PixelFormatDescriptor object. This value is 0x00000000 if no pixel format is set.

**bOpenGL (4 bytes):** An unsigned integer that indicates whether OpenGL commands are present in the metafile.

| Value | Meaning |
| --- | --- |
| 0x00000000 | OpenGL records are not present in the metafile. |
| 0x00000001 | OpenGL records are present in the metafile. |

### HeaderExtension2 Object

The HeaderExtension2 object defines the second extension to the [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7) header. It adds the ability to measure device surfaces in micrometers, which enhances the resolution and scalability of EMF metafiles.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| MicrometersX | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MicrometersY | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**MicrometersX (4 bytes):** The 32-bit horizontal size of the display device for which the metafile image was generated, in micrometers.

**MicrometersY (4 bytes):** The 32-bit vertical size of the display device for which the metafile image was generated, in micrometers.

### LogBrushEx Object

The **LogBrushEx** object defines the style, color, and pattern of a device-independent brush.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| BrushStyle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Color | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BrushHatch | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**BrushStyle (4 bytes):** An unsigned integer that specifies the brush style. The value MUST be an enumeration from BrushStyle enumeration ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.1.1.4). The style values that are supported in this structure are listed later in this section. The BS\_NULL style SHOULD be used to specify a brush that has no effect.

**Color (4 bytes):** A 32-bit ColorRef object ([MS-WMF] section 2.2.2.8) that specifies a color. The interpretation of this field depends on the value of **BrushStyle**, as explained in the following table.

**BrushHatch (4 bytes):** A 32-bit unsigned field that contains the brush hatch data. Its interpretation depends on the value of **BrushStyle**, as explained in the following table.

The following table shows the relationship between the **BrushStyle**, **Color**, and **BrushHatch** fields in a LogBrushEx object. Only supported brush styles are listed.

| BrushStyle | Color | BrushHatch |
| --- | --- | --- |
| **BS\_SOLID** | A ColorRef object, which specifies the color of the brush. | Not used and SHOULD be ignored. |
| **BS\_NULL** | Not used and SHOULD be ignored. | Not used and SHOULD be ignored. |
| **BS\_HATCHED** | A ColorRef object, which specifies the foreground color of the hatch pattern. | A value from the HatchStyle enumeration (section [2.1.17](#Section_db3f9395eb7540a7ba9f0f836658712d)), which specifies the orientation of lines used to create the hatch. |

### LogFont Object

The LogFont object specifies the basic attributes of a logical font.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Height | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Width | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Escapement | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Orientation | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Weight | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Italic | | | | | | | | Underline | | | | | | | | StrikeOut | | | | | | | | CharSet | | | | | | | |
| OutPrecision | | | | | | | | ClipPrecision | | | | | | | | Quality | | | | | | | | PitchAndFamily | | | | | | | |
| Facename (64 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Height (4 bytes):** A signed integer that specifies the height of the font's character cell in logical units. The character height value, also known as the [**em size**](#gt_d27c8bde-0ac3-4410-bc49-5b988ab28dd6), is the character [**cell height**](#gt_b97118e3-2175-4af5-b1f9-462e57501090) value minus the [**internal leading**](#gt_9efb809f-982e-4f1e-9dc8-89a65f22c732) value. The [**font mapper**](#gt_3c6f410b-e1f3-4841-8254-6f45433425f1) SHOULD interpret the value specified in the **Height** field in the following manner.

| Value | Meaning |
| --- | --- |
| 0x00000000 < *value* | The font mapper transforms this value into device units and matches it against the cell height of the available fonts. |
| 0x00000000 | The font mapper uses a default height value when it searches for a match. |
| *value* < 0x00000000 | The font mapper transforms this value into device units and matches its absolute value against the character height of the available fonts. |

For all height comparisons, the font mapper SHOULD look for the largest font that does not exceed the requested size.

**Width (4 bytes):** A signed integer that specifies the average width of characters in the font in logical units. If the **Width** field value is zero, an appropriate value SHOULD[<46>](#Appendix_A_46" \o "Product behavior note 46) be calculated from other values in this object to find a font that has the typographer's intended [**aspect ratio**](#gt_091ff8ed-6ac1-4015-9d91-93020f13f6b5).

**Escapement (4 bytes):** A signed integer that specifies the angle, in tenths of degrees, between the escapement vector and the x-axis of the device. The escapement vector is parallel to the [**baseline**](#gt_81ddcdfb-c66c-4c83-b963-18da9456fa62) of a row of text.

When the graphics mode is set to **GM\_ADVANCED**, the escapement angle of the string can be specified independently of the orientation angle of the string's characters. Graphics modes are specified in section [2.1.16](#Section_fe60a5a523044b639949e11bfdbebae9)

**Orientation (4 bytes):** A signed integer that specifies the angle, in tenths of degrees, between each character's baseline and the x-axis of the device.

**Weight (4 bytes):** A signed integer that specifies the [**weight**](#gt_fc6de273-4e8b-4f97-89d0-3dc1b4be9e8d) of the font in the range zero through 1000. For example, 400 is normal and 700 is bold. If this value is zero, a default weight can be used.[<47>](#Appendix_A_47" \o "Product behavior note 47)

**Italic (1 byte):** An unsigned integer that specifies an italic font if set to 0x01; otherwise, it MUST be set to 0x00.

**Underline (1 byte):** An unsigned integer that specifies an underlined font if set to 0x01; otherwise, it MUST be set to 0x00.

**StrikeOut (1 byte):** An unsigned integer that specifies a strikeout font if set to 0x01; otherwise, it MUST be set to 0x00.

**CharSet (1 byte):** An unsigned integer that specifies the set of character glyphs. It MUST be a value in the CharacterSet enumeration ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.1.1.5). If the character set is unknown, [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7) processing SHOULD NOT attempt to translate or interpret strings that are rendered with that font.

If a [**typeface**](#gt_2fff9b0b-4bc1-4d05-81ef-8dde59fd9789) name is specified in the **Facename** field, the **CharSet** field value MUST match the character set of that typeface.

**OutPrecision (1 byte):** An unsigned integer that specifies the output precision. The output precision defines how closely the font is required to match the requested height, width, character orientation, escapement, [**pitch**](#gt_0f2ba14a-c9b3-4cc8-8840-a24c0aed5eb9), and font type. It MUST be a value from the OutPrecision enumeration ([MS-WMF] section 2.1.1.21).

Applications can use the output precision to control how the font mapper chooses a font when the operating system contains more than one font with a specified name. For example, if an operating system contains a font named **Symbol** in [**rasterized**](#gt_e4616703-011e-4bcf-842a-fbd0f778d76c) and [**TrueType**](#gt_855615c9-5378-459f-8d58-20f640afed6e) forms, an output precision value of **OUT\_TT\_PRECIS** forces the font mapper to choose the TrueType version. A value of **OUT\_TT\_ONLY\_PRECIS** forces the font mapper to choose a TrueType font, even if it is necessary to substitute a TrueType font with another name.

**ClipPrecision (1 byte):** An unsigned integer that specifies the clipping precision. The clipping precision defines how to clip characters that are partially outside the clipping [**region**](#gt_6f13fe5f-8747-4ae0-9375-814bf0528197). It can be one or more of the ClipPrecision Flags ([MS-WMF] section 2.1.2.1).

**Quality (1 byte):** An unsigned integer that specifies the output quality. The output quality defines how closely to attempt to match the logical-font attributes to those of an actual physical font. It MUST be one of the values in the FontQuality enumeration ([MS-WMF] section 2.1.1.10).

**PitchAndFamily (1 byte):** A PitchAndFamily object ([MS-WMF] section 2.2.2.14) that specifies the pitch and family of the font. Font families describe the look of a font in a general way. They are intended for specifying a font when the specified typeface is not available.

**Facename (64 bytes):** A string of no more than 32 [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) characters that specifies the typeface name of the font. If the length of this string is less than 32 characters, a terminating NULL MUST be present, after which the remainder of this field MUST be ignored.

### LogFontEx Object

The LogFontEx object specifies the extended attributes of a logical font.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| LogFont (92 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| FullName (128 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Style (64 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Script (64 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**LogFont (92 bytes):** A [LogFont (section 2.2.13)](#Section_1cd7cb7a1a7c48d2b4a4c218f9d12100) object that specifies the basic attributes of the logical font.

**FullName (128 bytes):** A string of 64 [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) characters that contains the font's full name. If the length of this string is less than 64 characters, a terminating NULL MUST be present, after which the remainder of this field MUST be ignored.

**Style (64 bytes):** A string of 32 Unicode characters that defines the font's style. If the length of this string is less than 32 characters, a terminating NULL MUST be present, after which the remainder of this field MUST be ignored.

**Script (64 bytes):** A string of 32 Unicode characters that defines the character set of the font. If the length of this string is less than 32 characters, a terminating NULL MUST be present, after which the remainder of this field MUST be ignored.

### LogFontExDv Object

The LogFontExDv object specifies the [**design vector**](#gt_68697457-d046-45a9-b822-86cb2b8c3320) for an extended logical font.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| LogFontEx (348 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DesignVector (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**LogFontEx (348 bytes):** A [LogFontEx object (section 2.2.14)](#Section_71516231003445f2af78c463b98db4af) that specifies the extended attributes of the logical font.

**DesignVector (variable):** A [DesignVector object (section 2.2.3)](#Section_cdf6681fbf454eec923e08c8be16aaf2). This field MUST NOT be longer than 72 bytes.

A design vector SHOULD be specified only for a [**multiple master**](#gt_fef5ee7a-ad0c-4395-a1ac-f130405eec03) [**OpenType**](#gt_5e824598-05ae-44e2-99bd-78c347279ad9) font.

### LogFontPanose Object

The LogFontPanose object specifies the [**PANOSE**](#gt_8290d13f-8670-4b8e-ae6e-1ebb49319792) characteristics of a logical font.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| LogFont (92 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| FullName (128 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Style (64 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Version | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| StyleSize | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Match | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| VendorId | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Culture | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Panose | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | Padding | | | | | | | | | | | | | | | |

**LogFont (92 bytes):** A [LogFont (section 2.2.13)](#Section_1cd7cb7a1a7c48d2b4a4c218f9d12100) object that specifies the basic attributes of the logical font.

**FullName (128 bytes):** A string of 64 [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) characters that defines the font's full name. If the length of this string is less than 64 characters, a terminating NULL MUST be present, after which the remainder of this field MUST be ignored.

**Style (64 bytes):** A string of 32 Unicode characters that defines the font's style. If the length of this string is less than 32 characters, a terminating NULL MUST be present, after which the remainder of this field MUST be ignored.

**Version (4 bytes):** This field MUST be ignored.

**StyleSize (4 bytes):** An unsigned integer that specifies the point size at which [**font hinting**](#gt_d77e65f2-e12f-4d0f-866b-7252e368b32f) is performed. If set to zero, font hinting is performed at the point size corresponding to the **Height** field in the LogFont object in the **LogFont** field.

**Match (4 bytes):** This field MUST be ignored.

**Reserved (4 bytes):** An unsigned integer that MUST be set to zero and MUST be ignored.

**VendorId (4 bytes):** This field MUST be ignored.

**Culture (4 bytes):** An unsigned integer that MUST be set to zero and MUST be ignored.

**Panose (10 bytes):** A [Panose object (section 2.2.21)](#Section_4f1d246221334247ae1b2192cd91adb6) that specifies the PANOSE characteristics of the logical font. If all fields of this object are zero, it MUST be ignored.

**Padding (2 bytes):** A field that exists only to ensure 32-bit alignment of this structure. It MUST be ignored.

### LogPalette Object

The **LogPalette** object specifies a [**logical\_palette**](#gt_43e451cc-1df4-4abc-bb24-aac32f5ab6f3) that contains device-independent color definitions.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Version | | | | | | | | | | | | | | | | NumberOfEntries | | | | | | | | | | | | | | | |
| PaletteEntries (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Version (2 bytes):** An unsigned integer that specifies the version number of the system. This value is 0x0300.

**NumberOfEntries (2 bytes):** An unsigned integer that specifies the number of entries in the **PaletteEntries** field.

**PaletteEntries (variable):** An array of LogPaletteEntry objects (section [2.2.18](#Section_c1f7b285be164112a4e60b2fd4c1d148)) that defines the color and usage of each entry in the logical\_palette.

[**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) MUST define colors as device-independent values because the [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7) itself is device-independent.

### LogPaletteEntry Object

The **LogPaletteEntry** object defines the values that make up a single entry in a LogPalette object (section [2.2.17](#Section_758f047d765e424c9204a833b7b4e527)).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Reserved | | | | | | | | Blue | | | | | | | | Green | | | | | | | | Red | | | | | | | |

**Reserved (1 byte):** An unsigned integer that MUST NOT be used and MUST be ignored.

**Blue (1 byte):** An unsigned integer that defines the blue [**intensity**](#gt_f44dd201-e056-4467-bf8a-2dc1f4586a44) value for the entry.

**Green (1 byte):** An unsigned integer that defines the green intensity value for the entry.

**Red (1 byte):** An unsigned integer that defines the red intensity value for the entry.

[**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) MUST define colors as device-independent values because the [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7) itself is device-independent.

### LogPen Object

The **LogPen** object defines the style, width, and color of a logical pen.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| PenStyle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Width | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ColorRef | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**PenStyle (4 bytes):** An unsigned integer that specifies a value from the PenStyle enumeration (section [2.1.25](#Section_b2e629158401429fbb6db2506a4c2fe8)).

**Width (8 bytes):** A PointL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.15) that specifies the width of the pen by the value of its **x** field. The value of its **y** field MUST be ignored.

If the pen type in the **PenStyle** field is PS\_GEOMETRIC, this value is the width in logical units; otherwise, the width is specified in device units. If the pen type in the **PenStyle** field is PS\_COSMETIC, this value MUST be 0x00000001.

**ColorRef (4 bytes):** A **ColorRef** object ([MS-WMF] section 2.2.2.8) that specifies the pen color value.

### LogPenEx Object

The **LogPenEx** object specifies the style, width, and color of an extended logical pen.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| PenStyle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Width | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BrushStyle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ColorRef | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BrushHatch | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NumStyleEntries | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| StyleEntry (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**PenStyle (4 bytes):** An unsigned integer that specifies the pen style. This value is defined from the PenStyle enumeration (section [2.1.25](#Section_b2e629158401429fbb6db2506a4c2fe8)).

The pen style is a combination of pen type, line style, [**line cap**](#gt_b0341150-ca76-4424-88fe-607e03848dbf), and [**line join**](#gt_4fd3ebb8-54f8-47e8-aded-399a08ad873e).

**Width (4 bytes):** An unsigned integer that specifies the width of the line drawn by the pen.

If the pen type in the **PenStyle** field is PS\_GEOMETRIC, this value is the width in logical units; otherwise, the width is specified in device units. If the pen type in the **PenStyle** field is PS\_COSMETIC, this value MUST be 0x00000001.

**BrushStyle (4 bytes):** An unsigned integer that specifies a brush style for the pen from the BrushStyle enumeration ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.1.1.4).

If the pen type in the **PenStyle** field is PS\_GEOMETRIC, this value is either BS\_SOLID or BS\_HATCHED. The value of this field can be BS\_NULL, but only if the line style specified in **PenStyle** is PS\_NULL. The BS\_NULL style SHOULD be used to specify a brush that has no effect.

**ColorRef (4 bytes):** A ColorRef object ([MS-WMF] section 2.2.2.8). The interpretation of this field depends on the **BrushStyle** value, as shown in the table later in this section.

**BrushHatch (4 bytes):** The brush hatch pattern. The definition of this field depends on the **BrushStyle** value, as shown in the table later in this section.

**NumStyleEntries (4 bytes):** The number of elements in the array specified in the **StyleEntry** field. This value SHOULD be zero if **PenStyle** does not specify PS\_USERSTYLE.

**StyleEntry (variable, optional):** An array of 32-bit unsigned integers that defines the lengths of dashes and gaps in the line drawn by this pen when the value of **PenStyle** is PS\_USERSTYLE. The array contains the number of entries specified by **NumStyleEntries**, but it is used as if it repeated indefinitely.

The first entry in the **StyleEntry** array specifies the length of the first dash. The second entry specifies the length of the first gap. Thereafter, lengths of dashes and gaps alternate.

If the pen type in the **PenStyle** field is PS\_GEOMETRIC, lengths are specified in logical units; otherwise, they are specified in device units.

The LogPenEx object includes the specification of brush attributes, so it can be used to draw lines that consist of custom or predefined patterns. The following table shows the relationship between the **BrushStyle**, **ColorRef**, and **BrushHatch** fields in this object. Only supported brush styles are listed.

| BrushStyle | ColorRef | BrushHatch |
| --- | --- | --- |
| **BS\_SOLID** | A ColorRef object that specifies the color of lines drawn by the pen. | Not used and is ignored. |
| **BS\_NULL** | Not used and is ignored. | Not used and is ignored. |
| **BS\_HATCHED** | A ColorRef object that specifies the foreground color of the hatch pattern. | A value from the HatchStyle enumeration (section [2.1.17](#Section_db3f9395eb7540a7ba9f0f836658712d)) that specifies the orientation of lines used to create the hatch. If **PS\_GEOMETRIC** is not set in the **PenStyle** field, this field MUST be either **HS\_SOLIDTEXTCLR (0x0008)** or **HS\_SOLIDBKCLR (0x000A)**. |
| **BS\_PATTERN** | The low-order 16-bits is a value from the ColorUsage enumeration ([MS-WMF] section 2.1.1.6). | Not used and is ignored. |
| **BS\_DIBPATTERN** | The low-order 16 bits is a value from the ColorUsage enumeration. | Not used and is ignored. |
| **BS\_DIBPATTERNPT** | The low-order word is be a value from the ColorUsage enumeration. | Not used and is ignored. |

### Panose Object

The Panose object describes the PANOSE font-classification values for a TrueType font. These characteristics are used to associate the font with other fonts of similar appearance but different names.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| FamilyType | | | | | | | | SerifStyle | | | | | | | | Weight | | | | | | | | Proportion | | | | | | | |
| Contrast | | | | | | | | StrokeVariation | | | | | | | | ArmStyle | | | | | | | | Letterform | | | | | | | |
| Midline | | | | | | | | XHeight | | | | | | | |

**FamilyType (1 byte):** An unsigned integer that specifies the family type. This value is in the [FamilyType (section 2.1.12)](#Section_cfb9abd0a0a446dfb782eb856e19a5d3) enumeration table.

**SerifStyle (1 byte):** An unsigned integer that specifies the serif style. This value is in the [SerifType (section 2.1.30)](#Section_750aae5cb2574e15acfac1be27b8c408) enumeration table.

**Weight (1 byte):** An unsigned integer that specifies the [**weight**](#gt_fc6de273-4e8b-4f97-89d0-3dc1b4be9e8d) of the font. This value is in the [Weight (section 2.1.34)](#Section_46caba8a83e5450d8c4dd3999be38b93) enumeration table.

**Proportion (1 byte):** An unsigned integer that specifies the proportion of the font. This value is in the [Proportion (section 2.1.28)](#Section_d59152c4deb049d88eeb0407167698fe) enumeration table.

**Contrast (1 byte):** An unsigned integer that specifies the contrast of the font. This value is in the [Contrast (section 2.1.8)](#Section_652d38bb5626462caf8ba5dcb828d814) enumeration table.

**StrokeVariation (1 byte):** An unsigned integer that specifies the stroke variation for the font. This value is in the [StrokeVariation (section 2.1.33)](#Section_08b1f3afc0c14ebf823a757ade94df22) enumeration table.

**ArmStyle (1 byte):** An unsigned integer that specifies the arm style of the font. This value is in the [ArmStyle (section 2.1.3)](#Section_3ae0077a0c9d4e26bc2d773858bb0ac3) enumeration table.

**Letterform (1 byte):** An unsigned integer that specifies the letterform of the font. This value is in the [Letterform (section 2.1.20)](#Section_ee9127c4596f4744a59f44d9c3c5760f) enumeration table.

**Midline (1 byte):** An unsigned integer that specifies the midline of the font. This value is in the [MidLine (section 2.1.23)](#Section_27eb84dc85b14a9dbf9a15808b2c61d8) enumeration table.

**XHeight (1 byte):** An unsigned integer that specifies the x height of the font. This value is in the [XHeight (section 2.1.35)](#Section_47e9daaf0fff4ea09b4fc30329c79b64) enumeration table.

### PixelFormatDescriptor Object

The **PixelFormatDescriptor** object specifies the pixel format of a drawing surface.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| nSize | | | | | | | | | | | | | | | | nVersion | | | | | | | | | | | | | | | |
| dwFlags | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| iPixelType | | | | | | | | cColorBits | | | | | | | | cRedBits | | | | | | | | cRedShift | | | | | | | |
| cGreenBits | | | | | | | | cGreenShift | | | | | | | | cBlueBits | | | | | | | | cBlueShift | | | | | | | |
| cAlphaBits | | | | | | | | cAlphaShift | | | | | | | | cAccumBits | | | | | | | | cAccumRedBits | | | | | | | |
| cAccumGreenBits | | | | | | | | cAccumBlueBits | | | | | | | | cAccumAlphaBits | | | | | | | | cDepthBits | | | | | | | |
| cStencilBits | | | | | | | | cAuxBuffers | | | | | | | | iLayerType | | | | | | | | bReserved | | | | | | | |
| dwLayerMask | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwVisibleMask | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwDamageMask | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**nSize (2 bytes):** An unsigned integer that specifies the size in bytes, of this data structure.

**nVersion (2 bytes):** An unsigned integer that MUST be set to 0x0001.

**dwFlags (4 bytes):** A set of bit flags that specify properties of the pixel buffer that is used for output to the drawing surface. These properties are not all mutually exclusive; combinations of flags are allowed, except where noted otherwise.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| P | F | S O | G | M | W | S | D | C | D A | D S | A | S L | S C | S E | S P | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | S D | D D | D P | 0 |

The following bit flag constants are defined.

| Value | Description |
| --- | --- |
| D  PFD\_DOUBLEBUFFER | The pixel buffer is double-buffered. This flag and **PFD\_SUPPORT\_GDI** MUST NOT both be set. |
| S  PFD\_STEREO | The pixel buffer MAY be [**stereoscopic**](#gt_c4dec1f3-42e3-4701-b49d-ca48115e1169); that is, it MAY specify a [**color plane**](#gt_78d62be3-d07a-447e-ab51-fa8be5560fe0) that is used to create the illusion of depth in an image.[<48>](#Appendix_A_48" \o "Product behavior note 48) |
| W  PFD\_DRAW\_TO\_WINDOW | The pixel buffer can draw to a window or device surface. |
| M  PFD\_DRAW\_TO\_BITMAP | The pixel buffer can draw to a memory [**bitmap**](#gt_26a17e58-eeba-4210-81b9-8bbadf7ee952). |
| G  PFD\_SUPPORT\_GDI | This flag SHOULD be clear, but it MAY be set. [<49>](#Appendix_A_49" \o "Product behavior note 49) The **PFD\_SUPPORT\_GDI** flag and **PFD\_DOUBLEBUFFER** MUST NOT both be set. |
| SO  PFD\_SUPPORT\_OPENGL | The pixel buffer supports [**OpenGL**](#gt_216becb9-a800-4a44-8206-2ec2231c9c67) [[OPENGL]](https://go.microsoft.com/fwlink/?LinkId=90241) drawing. |
| F  PFD\_GENERIC\_FORMAT | The pixel format is natively supported by the operating system; this is known as the "generic" implementation.[<50>](#Appendix_A_50" \o "Product behavior note 50)  If clear, the pixel format is supported by a device driver or hardware. |
| P  PFD\_NEED\_PALETTE | The buffer uses [**RGBA**](#gt_86add9f3-4083-49ce-a1b3-359595c14b65) pixels on a palette-managed device. A LogPalette object (section [2.2.17](#Section_758f047d765e424c9204a833b7b4e527)) is required to achieve the best results for this pixel type. Colors in the [**palette**](#gt_dca5cd91-8d01-424b-9722-c9e818562ee5) SHOULD be specified according to the values of the **cRedBits**, **cRedShift**, **cGreenBits**, **cGreenShift**, **cBlueBits**, and **cBlueShift** fields. |
| SP  PFD\_NEED\_SYSTEM\_PALETTE | The output device supports one hardware palette in 256-color mode only. For such systems to use hardware acceleration, the hardware palette MUST be in a fixed order (for example, 3-3-2) when in RGBA mode, or MUST match the LogPalette object when in [**color table**](#gt_4737483c-820c-41e2-a7be-0ffed2eb4418) mode. |
| SE  PFD\_SWAP\_EXCHANGE | The contents of the back buffer have been exchanged with the contents of the front buffer in a double-buffered color plane. |
| SC  PFD\_SWAP\_COPY | The contents of the back buffer have been copied to the front buffer in a double-buffered color plane. The contents of the back buffer have not been affected. |
| SL  PFD\_SWAP\_LAYER\_BUFFERS | A device can swap individual color planes with pixel formats that include double-buffered overlay or underlay color planes. Otherwise all color planes are swapped together as a group. |
| A  PFD\_GENERIC\_ACCELERATED | The pixel format is supported by a device driver that accelerates the generic implementation. If this flag is clear and the **PFD\_GENERIC\_FORMAT** flag is set, the pixel format is supported by the generic implementation only. |
| DS  PFD\_SUPPORT\_DIRECTDRAW | The pixel buffer supports DirectDraw drawing, which allows applications to have low-level control of the output drawing surface. |
| DA  PFD\_DIRECT3D\_ACCELERATED | The pixel buffer supports Direct3D drawing, which accellerated rendering in three dimensions. |
| C  PFD\_SUPPORT\_COMPOSITION | The pixel buffer supports [**compositing**](#gt_95602299-4a8f-4cee-8497-d9500091489c), which indicates that source pixels MAY overwrite or be combined with background pixels.[<51>](#Appendix_A_51" \o "Product behavior note 51) |
| DP  PFD\_DEPTH\_DONTCARE | The pixel buffer is not required to include a color plane for depth effects. |
| DD  PFD\_DOUBLEBUFFER\_DONTCARE | The pixel buffer can be either single or double buffered. |
| SD  PFD\_STEREO\_DONTCARE | The pixel buffer MAY be either [**monoscopic**](#gt_b1bf1a3d-8fed-4357-8557-456676c7416b) or stereoscopic. |

**iPixelType (1 byte):** The type of pixel data.

| Value | Meaning |
| --- | --- |
| PFD\_TYPE\_RGBA  0x00 | The pixel format is RGBA. |
| PFD\_TYPE\_COLORINDEX  0x01 | Each pixel is an index in a color table. |

**cColorBits (1 byte):** The number of bits per pixel for RGBA pixel types, excluding the [**alpha**](#gt_134d5688-b046-4a5b-a74e-c0c11fc04eb3) bitplanes. For color table pixels, it is the size of each color table index.

**cRedBits (1 byte):** Specifies the number of red bitplanes in each RGBA color buffer.

**cRedShift (1 byte):** Specifies the shift count in bits for red bitplanes in each RGBA color buffer.

**cGreenBits (1 byte):** Specifies the number of green bitplanes in each RGBA color buffer.

**cGreenShift (1 byte):** Specifies the shift count for green bitplanes in each RGBA color buffer.

**cBlueBits (1 byte):** Specifies the number of blue bitplanes in each RGBA color buffer.

**cBlueShift (1 byte):** Specifies the shift count for blue bitplanes in each RGBA color buffer.

**cAlphaBits (1 byte):** Specifies the number of alpha bitplanes in each RGBA color buffer.[<52>](#Appendix_A_52" \o "Product behavior note 52)

**cAlphaShift (1 byte):** Specifies the shift count for alpha bitplanes in each RGBA color buffer.[<53>](#Appendix_A_53" \o "Product behavior note 53)

**cAccumBits (1 byte):** Specifies the total number of bitplanes in the accumulation buffer.

**cAccumRedBits (1 byte):** Specifies the number of red bitplanes in the accumulation buffer.

**cAccumGreenBits (1 byte):** Specifies the number of green bitplanes in the accumulation buffer.

**cAccumBlueBits (1 byte):** Specifies the number of blue bitplanes in the accumulation buffer.

**cAccumAlphaBits (1 byte):** Specifies the number of alpha bitplanes in the accumulation buffer.[<54>](#Appendix_A_54" \o "Product behavior note 54)

**cDepthBits (1 byte):** Specifies the depth of the depth (z-axis) buffer.

**cStencilBits (1 byte):** Specifies the depth of the stencil buffer.

**cAuxBuffers (1 byte):** Specifies the number of auxiliary buffers. Auxiliary buffers are not supported.

**iLayerType (1 byte):** This field MAY be ignored.

**bReserved (1 byte):** Specifies the number of overlay and underlay planes. Bits 0 through 3 specify up to 15 overlay planes and bits 4 through 7 specify up to 15 underlay planes.

**dwLayerMask (4 bytes):** This field MAY be ignored.

**dwVisibleMask (4 bytes):** Specifies the transparent color or index of an underlay plane. When the pixel type is RGBA, **dwVisibleMask** is a transparent [**RGB**](#gt_2c716d3a-e60b-4e52-bbb0-2fdeb298003b) color value. When the pixel type is color index, it is a transparent index value.

**dwDamageMask (4 bytes):** This field SHOULD be ignored.

The PixelFormatDescriptor object is used in EMR\_HEADER records (section [2.3.4.2](#Section_de081cd7351f4cc2830bd03fb55e89ab)) to specify the pixel format of the output surface.

### Point28\_4 Object

The Point28\_4 object represents the location of a point on a device surface with coordinates in [**28.4 bit FIX notation**](#gt_1bcc53a6-1861-49e9-bd00-3a38a888d9f4).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| x | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| y | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**x (4 bytes):** A [BitFIX28\_4 object (section 2.2.1)](#Section_075e694f901848ada22c81dd07c67c69) that represents the horizontal coordinate of the point.

**y (4 bytes):** A BitFIX28\_4 object that represents the vertical coordinate of the point.

### RegionData Object

The **RegionData** object specifies data that defines a region, which is made of non-overlapping rectangles.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| RegionDataHeader (32 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**RegionDataHeader (32 bytes):** A 256-bit RegionDataHeader object (section [2.2.25](#Section_5ca68a15181145b6a51b5e40d1055ccc)) that defines the contents of the **Data** field.

**Data (variable):** An array of RectL objects ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19); the objects are merged to create the [**region**](#gt_6f13fe5f-8747-4ae0-9375-814bf0528197).

### RegionDataHeader Object

The **RegionDataHeader** object defines the properties of a RegionData (section [2.2.24](#Section_e66601f29b5c46198476ddb7b087551b)) object.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CountRects | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| RgnSize | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Size (4 bytes):** An unsigned integer that specifies the size of this object in bytes. This value is 0x00000020.

**Type (4 bytes):** An unsigned integer that specifies the [**region**](#gt_6f13fe5f-8747-4ae0-9375-814bf0528197) type. This value is 0x00000001.

**CountRects (4 bytes):** An unsigned integer that specifies the number of rectangles in this region.

**RgnSize (4 bytes):** An unsigned integer that specifies the size of the buffer of rectangles in bytes.

**Bounds (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19), which specifies the bounds of the region.

### TriVertex Object

The TriVertex object specifies color and position information for the definition of a rectangle or triangle vertex.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| x | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| y | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Red | | | | | | | | | | | | | | | | Green | | | | | | | | | | | | | | | |
| Blue | | | | | | | | | | | | | | | | Alpha | | | | | | | | | | | | | | | |

**x (4 bytes):** A signed integer that specifies the horizontal position, in logical units.

**y (4 bytes):** A signed integer that specifies the vertical position, in logical units.

**Red (2 bytes):** An unsigned integer that specifies the red color value for the point.

**Green (2 bytes):** An unsigned integer that specifies the green color value for the point.

**Blue (2 bytes):** An unsigned integer that specifies the blue color value for the point.

**Alpha (2 bytes):** An unsigned integer that specifies the [**alpha transparency**](#gt_134d5688-b046-4a5b-a74e-c0c11fc04eb3) value for the point.

### UniversalFontId Object

The **UniversalFontId** object defines a mechanism for identifying fonts in [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) [**metafiles**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Checksum | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Index | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Checksum (4 bytes):** An unsigned integer that is the checksum of the font. The checksum value has the following meanings.

| Value | Meaning |
| --- | --- |
| 0x00000000 | The object is a device font. |
| 0x00000001 | The object is a [**Type 1 font**](#gt_8539bd4b-4321-40c3-b4d2-aefe3c846e4e) that has been installed on the client machine and is enumerated by the [**PostScript**](#gt_31971c63-f974-47de-8de9-6b76d093890d) [**printer driver**](#gt_1a48eebd-e72c-494d-b8cb-84dfb7bc3b65) as a device font. |
| 0x00000002 | The object is not a font but is a Type 1 [**rasterizer**](#gt_62128f61-b612-4197-a74c-2ad6a7cf9206). |
| 3 ≤ *value* | The object is a [**bitmap**](#gt_26a17e58-eeba-4210-81b9-8bbadf7ee952), vector, or [**TrueType**](#gt_855615c9-5378-459f-8d58-20f640afed6e) font, or a Type 1 [**rasterized font**](#gt_e4616703-011e-4bcf-842a-fbd0f778d76c) that was created by a Type 1 rasterizer.  A checksum value SHOULD be computed for the font and compared to the value in this field. If it matches, it is considered to be the same as the font referenced by this metafile record. If it does not match, the system [**font mapper**](#gt_3c6f410b-e1f3-4841-8254-6f45433425f1) MAY use a default mechanism to select a back-up font.[<55>](#Appendix_A_55" \o "Product behavior note 55) |

If a checksum value is computed, it SHOULD be computed using the following algorithm.

For the purpose of this computation, the font is considered simply to be a stream of bytes that is external to this [EMF record](#Section_e0137630f3ad492cbde9e68866e255ba). Any larger file structure in which the font might reside is system-dependent or implementation-dependent.

1. ULONG ComputeFileviewCheckSum(PVOID pvView, ULONG cjView)
2. {
3. ULONG sum;
4. PULONG pulCur,pulEnd;
5. pulCur = (PULONG) pvView;
6. for (sum = 0, pulEnd = pulCur + cjView / sizeof(ULONG);
7. pulCur < pulEnd; pulCur += 1)
8. {
9. sum += 256 \* sum + \*pulCur;
10. }
11. return ( sum < 2 ) ? 2 : sum;
12. }

**pvView**: A pointer to the start of the font.

**cjView**: The length of the font in bytes.

**Index (4 bytes):** An unsigned integer that is an index associated with the font object. The meaning of this field is determined by the type of font.

### XForm Object

The **XForm** object defines a two-dimensional, linear [**transform**](#gt_fdca4a96-8e04-4dc0-acad-1d01e1384a5b) matrix.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| M11 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| M12 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| M21 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| M22 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Dx | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Dy | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**M11 (4 bytes):** A FLOAT matrix value.

**M12 (4 bytes):** A FLOAT matrix value.

**M21 (4 bytes):** A FLOAT matrix value.

**M22 (4 bytes):** A FLOAT matrix value.

**Dx (4 bytes):** A FLOAT value that contains a horizontal [**translation**](#gt_4d42abda-3b31-4828-bc4c-684f2de4e2af) component, in logical units.

**Dy (4 bytes):** A FLOAT value that contains a vertical translation component, in logical units.

The following equations specify how the matrix values are used to transform a point (X,Y) to a new point (X',Y'):

1. X' = M11 \* X + M21 \* Y + Dx
2. Y' = M12 \* X + M22 \* Y + Dy

For more information concerning transforms and [**coordinate spaces**](#gt_84184077-d0d9-41e0-acaf-6596aa066813), see [[MSDN-WRLDPGSPC]](https://go.microsoft.com/fwlink/?LinkId=90169).

## EMF Records

This section specifies the [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7) records, which have been grouped into the following categories.

| Name | Section | Description |
| --- | --- | --- |
| Bitmap record types | [2.3.1](#Section_2a57d4bdf3dc45df95fd447382b5262a) | Manage and output [**bitmap**](#gt_26a17e58-eeba-4210-81b9-8bbadf7ee952) images. |
| Clipping record types | [2.3.2](#Section_0ca0d18e324e452f9a4126e1a82e3e03) | Specify and manage clipping regions. |
| Comment record types | [2.3.3](#Section_05940d07e1124146ac0588fc6a1f70b9) | Define formats for specifying arbitrary private data, embedding records in other metafile formats, and adding new or special-purpose commands. |
| Control record types | [2.3.4](#Section_2caf0f4781a44079931dfca6fa499869) | Define the start and end of an EMF metafile and its properties. |
| Drawing record types | [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4) | Perform graphics drawing. |
| Escape record types | [2.3.6](#Section_5370873a5fe64fdd8c5abade4ee54a1f) | Execute [**printer driver**](#gt_1a48eebd-e72c-494d-b8cb-84dfb7bc3b65) functions. |
| Object creation record types | [2.3.7](#Section_7d914464d5464393bddb6d1e747fca18) | Create graphics objects. |
| Object manipulation record types | [2.3.8](#Section_2f1c3268b43b4c28988b41aa49354998) | Manage and modify graphics objects. |
| OpenGL record types | [2.3.9](#Section_4a6c2fc239af43b48254449f5f1c5c05) | Specify metafile records generated by [**OpenGL**](#gt_216becb9-a800-4a44-8206-2ec2231c9c67) [[OPENGL]](https://go.microsoft.com/fwlink/?LinkId=90241). |
| Path bracket record types | [2.3.10](#Section_b930b98930ec4954a8891dc60ce0b689) | Specify and manipulate paths in [**path brackets**](#gt_b89ea7da-c0bf-4c45-9f3e-fc430d7ab601). |
| State record types | [2.3.11](#Section_8f8a4df2f8d74d39afc094e19f524652) | Specify and manage graphics properties. |
| Transform record types | [2.3.12](#Section_a0442ad12d6e43e7ac8ce252e492f97e) | Specify and modify [**world-space**](#gt_baadb707-1343-413b-9b36-2496b06e9bd7) to [**page-space**](#gt_f530f123-858a-4af7-b626-adea0aeaa75a) [**transforms**](#gt_fdca4a96-8e04-4dc0-acad-1d01e1384a5b). |

All EMF records MUST be multiples of 4 bytes in length; hence, each record starts on a 32-bit offset from the start of the metafile. To ensure each subsequent record also starts on a 32-bit boundary, an **AlignmentPadding** field is used, if necessary. The contents of **AlignmentPadding** fields are indeterminate and MUST be ignored. In general, such fields are shown only in the generic definitions of record categories.

To ensure maximum compatibility, implementations need to allow for record truncation for unused fields at the end of an EMF record. If there are extra data at the end of the record undocumented by this specification, these data MUST be ignored.

### Bitmap Record Types

The **Bitmap** record types perform block transfers of [**bitmap**](#gt_26a17e58-eeba-4210-81b9-8bbadf7ee952) images.

The following are the bitmap record types.

| Name | Section | Description |
| --- | --- | --- |
| EMR\_ALPHABLEND | [2.3.1.1](#Section_34e07d4faee64b63a4bb96996ad47669) | Specifies a block transfer of pixels from a source bitmap to a destination rectangle, including [**alpha transparency**](#gt_134d5688-b046-4a5b-a74e-c0c11fc04eb3) data, according to a specified blending operation. |
| EMR\_BITBLT | [2.3.1.2](#Section_347d1c44184747ec87627059e9e9b185) | Specifies a block transfer of pixels from a source bitmap to a destination rectangle, optionally in combination with a brush pattern, according to a specified [**raster operation**](#gt_772681cd-b9a5-440c-8021-b6219c70f6fc). |
| EMR\_MASKBLT | [2.3.1.3](#Section_e6f715c0d0344eb3952ef8ee66adb9ed) | Specifies a block transfer of pixels from a source bitmap to a destination rectangle, optionally in combination with a brush pattern and with the application of a color mask bitmap, according to specified foreground and background raster operations. |
| EMR\_PLGBLT | [2.3.1.4](#Section_b6c8b39d42c74221ab7093f1e09b644e) | Specifies a block transfer of pixels from a source bitmap to a destination parallelogram, with the application of a color mask bitmap. |
| EMR\_SETDIBITSTODEVICE | [2.3.1.5](#Section_e8816cc635d243e68d88d69cd342372e) | Specifies a block transfer of pixels from specified scanlines of a source bitmap to a destination rectangle. |
| EMR\_STRETCHBLT | [2.3.1.6](#Section_0dd551f380f7485289c29ddba803a192) | Specifies a block transfer of pixels from a source bitmap to a destination rectangle, optionally in combination with a brush pattern, according to a specified raster operation, stretching or compressing the output to fit the dimensions of the destination, if necessary. |
| EMR\_STRETCHDIBITS | [2.3.1.7](#Section_89c0d8080dea413fbe402e9e51fa36ac) | Specifies a block transfer of pixels from a source bitmap to a destination rectangle, optionally in combination with a brush pattern, according to a specified raster operation, stretching or compressing the output to fit the dimensions of the destination, if necessary. |
| EMR\_TRANSPARENTBLT | [2.3.1.8](#Section_aa2160512dc04317b343525431cfa103) | Specifies a block transfer of pixels from a source bitmap to a destination rectangle, treating a specified color as transparent, stretching or compressing the output to fit the dimensions of the destination, if necessary. |

The generic structure of bitmap records is specified as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BitmapRecordBuffer (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that defines the type of record. The bitmap record types are listed in the following table. See the preceding table for descriptions of these record types.

| Name | Value |
| --- | --- |
| EMR\_BITBLT | 0x0000004C |
| EMR\_STRETCHBLT | 0x0000004D |
| EMR\_MASKBLT | 0x0000004E |
| EMR\_PLGBLT | 0x0000004F |
| EMR\_SETDIBITSTODEVICE | 0x00000050 |
| EMR\_STRETCHDIBITS | 0x00000051 |
| EMR\_ALPHABLEND | 0x00000072 |
| EMR\_TRANSPARENTBLT | 0x00000074 |

**Size (4 bytes):** An unsigned integer that specifies the size in bytes of this record in the [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7). This value MUST be a multiple of 4 bytes.

**BitmapRecordBuffer (variable):** An array of bytes that contains the remainder of the bitmap record.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| BitmapRecordParm (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| AlignmentPadding (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**BitmapRecordParm (variable):** An array of bytes that contains the parameters for the bitmap record.

**AlignmentPadding (variable, optional):** An array of up to 3 bytes that pads the record so that its total size is a multiple of 4 bytes. This field MUST be ignored.

The following notes apply generally to [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) bitmap block transfers, unless specified otherwise:

* Source and mask bitmaps are in [**DIB**](#gt_3a6934d5-cc59-4ab9-a54e-e062a09a4091) format. DIBs are specified by DeviceIndependentBitmap objects ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.9).
* The clipping [**regions**](#gt_6f13fe5f-8747-4ae0-9375-814bf0528197) used by bitmap records are maintained in a **Regions** state element (section [3.1.1.2.1](#Section_939400bc0ecd4a28aa8c60c7e4cc9877)) in the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45) (section [3.1](#Section_50ae16a09d1748018bf1e28e9533a889)).
* If the color format of the source or pattern bitmap does not match the color format of the destination, the source pr pattern bits MUST be converted to the destination format prior to performing the block transfer.
* If the source and destination rectangles are not the same size, the source bitmap MUST be expanded or compressed to match the destination rectangle. This stretching function is performed according to a property from the StretchMode enumeration (section [2.1.32](#Section_e72f20aa2e2d453f9e95eaef50b76e19)).
* If an XForm object (section [2.2.28](#Section_e84107e9bc2b4a1492345d173adc1b59)) is specified, a [**world-space**](#gt_baadb707-1343-413b-9b36-2496b06e9bd7) to [**page-space**](#gt_f530f123-858a-4af7-b626-adea0aeaa75a) [**transform**](#gt_fdca4a96-8e04-4dc0-acad-1d01e1384a5b) SHOULD be applied to the source bitmap. [**Scaling**](#gt_f3f5f638-16f6-4159-a3f1-9d845b1196f5), [**translation**](#gt_4d42abda-3b31-4828-bc4c-684f2de4e2af), and [**reflection transforms**](#gt_bbc1ce5c-0fd6-4fde-abb3-40525a95147d) SHOULD be supported, and [**rotation**](#gt_5cdd5e5b-fe3d-46e6-9019-50483463a76a) and [**shear transforms**](#gt_de5c1f5c-2ce7-4e79-9be9-a01b09175f5b) MAY be supported.

For more information concerning transforms and [**coordinate spaces**](#gt_84184077-d0d9-41e0-acaf-6596aa066813), see [[MSDN-WRLDPGSPC]](https://go.microsoft.com/fwlink/?LinkId=90169).

See section [2.3](#Section_e0137630f3ad492cbde9e68866e255ba) for more EMF record types.

#### EMR\_ALPHABLEND Record

The **EMR\_ALPHABLEND** record specifies a block transfer of pixels from a source [**bitmap**](#gt_26a17e58-eeba-4210-81b9-8bbadf7ee952) to a destination rectangle, including [**alpha transparency**](#gt_134d5688-b046-4a5b-a74e-c0c11fc04eb3) data, according to a specified blending operation.[<56>](#Appendix_A_56" \o "Product behavior note 56)

Fields not specified in this section are specified in section [2.3.1](#Section_2a57d4bdf3dc45df95fd447382b5262a).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| xDest | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| yDest | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cxDest | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cyDest | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BLENDFUNCTION | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| xSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ySrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| XformSrc (24 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BkColorSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| UsageSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offBmiSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbBmiSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offBitsSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbBitsSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cxSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cySrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BitmapBuffer (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_ALPHABLEND**. This value is 0x00000072.

**Bounds (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19) that specifies the destination bounding rectangle in logical coordinates. If the intersection of this rectangle with the current clipping [**regions**](#gt_6f13fe5f-8747-4ae0-9375-814bf0528197) (section [3.1.1.2.1](#Section_939400bc0ecd4a28aa8c60c7e4cc9877)) in the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45) (section [3.1](#Section_50ae16a09d1748018bf1e28e9533a889)) is empty, this record has no effect.

**xDest (4 bytes):** A signed integer that specifies the logical x-coordinate of the upper-left corner of the destination rectangle.

**yDest (4 bytes):** A signed integer that specifies the logical y-coordinate of the upper-left corner of the destination rectangle.

**cxDest (4 bytes):** A signed integer that specifies the logical width of the destination rectangle. This value MUST be greater than zero.

**cyDest (4 bytes):** A signed integer that specifies the logical height of the destination rectangle. This value MUST be greater than zero.

**BLENDFUNCTION (4 bytes):** A structure that specifies the blending operations for source and destination bitmaps.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| BlendOperation | | | | | | | | BlendFlags | | | | | | | | SrcConstantAlpha | | | | | | | | AlphaFormat | | | | | | | |

**BlendOperation (1 byte):** The blend operation code. The only source and destination blend operation that has been defined is 0x00, which specifies that the source bitmap MUST be combined with the destination bitmap based on the alpha transparency values of the source pixels. See the following equations for details.

**BlendFlags (1 byte):** This value is 0x00 and MUST be ignored.

**SrcConstantAlpha (1 byte):** An unsigned integer that specifies alpha transparency, which determines the blend of the source and destination bitmaps. This value MUST be used on the entire source bitmap. The minimum alpha transparency value, zero, corresponds to completely transparent; the maximum value, 0xFF, corresponds to completely opaque. In effect, a value of 0xFF specifies that the per-pixel alpha values determine the blend of the source and destination bitmaps. See the equations later in this section for details.

**AlphaFormat (1 byte):** A structure that specifies how source and destination pixels are interpreted with respect to alpha transparency.

| Value | Meaning |
| --- | --- |
| 0x00 | The pixels in the source bitmap do not specify alpha transparency. In this case, the **SrcConstantAlpha** value determines the blend of the source and destination bitmaps. Note that in the following equations **SrcConstantAlpha** is divided by 255, which produces a value in the range 0 to 1. |
| AC\_SRC\_ALPHA  0x01 | Indicates that the source bitmap is 32 bits-per-pixel and specifies an alpha transparency value for each pixel. |

**xSrc (4 bytes):** A signed integer that specifies the logical x-coordinate of the upper-left corner of the source rectangle.

**ySrc (4 bytes):** A signed integer that specifies the logical y-coordinate of the upper-left corner of the source rectangle.

**XformSrc (24 bytes):** An [XForm](#Section_e84107e9bc2b4a1492345d173adc1b59) object (section 2.2.28) that specifies a [**world-space**](#gt_baadb707-1343-413b-9b36-2496b06e9bd7) to [**page-space**](#gt_f530f123-858a-4af7-b626-adea0aeaa75a) [**transform**](#gt_fdca4a96-8e04-4dc0-acad-1d01e1384a5b) to apply to the source bitmap.

**BkColorSrc (4 bytes):** A ColorRef object ([MS-WMF] section 2.2.2.8) that specifies the background color of the source bitmap.

**UsageSrc (4 bytes):** An unsigned integer that specifies how to interpret values in the [**color table**](#gt_4737483c-820c-41e2-a7be-0ffed2eb4418) in the source bitmap header. This value is in the [DIBColors](#Section_a5e722e3891a4a67be1aed5a48a7fda1) enumeration (section 2.1.9).

**offBmiSrc (4 bytes):** An unsigned integer that specifies the offset in bytes from the start of this record to the source bitmap header in the **BitmapBuffer** field.

**cbBmiSrc (4 bytes):** An unsigned integer that specifies the size in bytes of the source bitmap header.

**offBitsSrc (4 bytes):** An unsigned integer that specifies the offset in bytes from the start of this record to the source bitmap bits in the **BitmapBuffer** field.

**cbBitsSrc (4 bytes):** An unsigned integer that specifies the size in bytes of the source bitmap bits.

**cxSrc (4 bytes):** A signed integer that specifies the logical width of the source rectangle. This value MUST be greater than zero.

**cySrc (4 bytes):** A signed integer that specifies the logical height of the source rectangle. This value MUST be greater than zero.

**BitmapBuffer (variable):** A buffer containing the source bitmap, which is not required to be contiguous with the fixed portion of the **EMR\_ALPHABLEND** record. Thus, fields in this buffer that are labeled "UndefinedSpace" are optional and MUST be ignored.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| UndefinedSpace1 (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BmiSrc (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| UndefinedSpace2 (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BitsSrc (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**BmiSrc (variable):** The source bitmap header.

**BitsSrc (variable):** The source bitmap bits.

The following equations show how destination pixels are computed from source pixels using **BLENDFUNCTION**. In the equations, "dst" refers to the destination bitmap, and "src" refers to the source bitmap. The color and transparency values of the source and destination pixels are denoted by "Red", "Green", "Blue", and "Alpha".

**Case I**: The **AlphaFormat** value is 0, which means the **SrcConstantAlpha** value MUST be used to blend the source and destination bitmaps, as follows.

1. dst.Red = src.Red \* (SrcConstantAlpha/255.0) +
2. dst.Red \* (1.0 - (SrcConstantAlpha/255.0))
4. dst.Green = src.Green \* (SrcConstantAlpha/255.0) +
5. dst.Green \* (1.0 - (SrcConstantAlpha/255.0))
7. dst.Blue = src.Blue \* (SrcConstantAlpha/255.0) +
8. dst.Blue \* (1.0 - (SrcConstantAlpha/255.0))

If the destination bitmap has an alpha channel, then it is blended as follows.

1. dst.Alpha = src.Alpha \* (SrcConstantAlpha/255.0) +
2. dst.Alpha \* (1.0 - (SrcConstantAlpha/255.0))

Note that if **SrcConstantAlpha** is 0xFF, these equations reduce to a simple source copy to the destination.

**Case II**: The **AlphaFormat** value is **AC\_SRC\_ALPHA**, which means the source pixels MUST be premultiplied by **SrcConstantAlpha**, and then the blend MUST be based on the per-pixel source alpha channel, as follows.

1. src.Red = src.Red \* (SrcConstantAlpha/255.0)
3. src.Green = src.Green \* (SrcConstantAlpha/255.0)
5. src.Blue = src.Blue \* (SrcConstantAlpha/255.0)
7. dst.Red = src.Red + (1.0 - (src.Alpha/255.0)) \* dst.Red
9. dst.Green = src.Green + (1.0 - (src.Alpha/255.0)) \* dst.Green
11. dst.Blue = src.Blue + (1.0 - (src.Alpha/255.0)) \* dst.Blue

If the destination bitmap has an alpha channel, it is blended as follows.

1. src.Alpha = src.Alpha \* (SrcConstantAlpha)/255.0)
3. dst.Alpha = src.Alpha + (1.0 - (src.Alpha/255.0)) \* dst.Alpha

If **SrcConstantAlpha** is 0xFF, there is in effect no premultiplication of the source values.

See section 2.3.1 for more bitmap record types.

#### EMR\_BITBLT Record

The EMR\_BITBLT record specifies a block transfer of pixels from a source [**bitmap**](#gt_26a17e58-eeba-4210-81b9-8bbadf7ee952) to a destination rectangle, optionally in combination with a brush pattern, according to a specified [**raster operation**](#gt_772681cd-b9a5-440c-8021-b6219c70f6fc).

Fields not specified in this section are specified in section [2.3.1](#Section_2a57d4bdf3dc45df95fd447382b5262a).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| xDest | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| yDest | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cxDest | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cyDest | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BitBltRasterOperation | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| xSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ySrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| XformSrc (24 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BkColorSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| UsageSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offBmiSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbBmiSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offBitsSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbBitsSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BitmapBuffer (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as EMR\_BITBLT. This value is 0x0000004C.

**Bounds (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19) that specifies the destination bounding rectangle in logical coordinates. If the intersection of this rectangle with the current clipping [**regions**](#gt_6f13fe5f-8747-4ae0-9375-814bf0528197) (section [3.1.1.2.1](#Section_939400bc0ecd4a28aa8c60c7e4cc9877)) in the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45) (section [3.1](#Section_50ae16a09d1748018bf1e28e9533a889)) is empty, this record has no effect.

**xDest (4 bytes):** A signed integer that specifies the logical x-coordinate of the upper-left corner of the destination rectangle.

**yDest (4 bytes):** A signed integer that specifies the logical y-coordinate of the upper-left corner of the destination rectangle.

**cxDest (4 bytes):** A signed integer that specifies the logical width of the source and destination rectangles.

**cyDest (4 bytes):** A signed integer that specifies the logical height of the source and destination rectangles.

**BitBltRasterOperation (4 bytes):** An unsigned integer that specifies the raster operation code. This code defines how the color data of the source rectangle is to be combined with the color data of the destination rectangle and optionally a brush pattern, to achieve the final color.

This value is in the Ternary Raster Operation enumeration ([MS-WMF] section 2.1.1.31).

**xSrc (4 bytes):** A signed integer that specifies the logical x-coordinate of the upper-left corner of the source rectangle.

**ySrc (4 bytes):** A signed integer that specifies the logical y-coordinate of the upper-left corner of the source rectangle.

**XformSrc (24 bytes):** An [XForm](#Section_e84107e9bc2b4a1492345d173adc1b59) object (section 2.2.28) that specifies a [**world-space**](#gt_baadb707-1343-413b-9b36-2496b06e9bd7) to [**page-space**](#gt_f530f123-858a-4af7-b626-adea0aeaa75a) [**transform**](#gt_fdca4a96-8e04-4dc0-acad-1d01e1384a5b) to apply to the source bitmap.

**BkColorSrc (4 bytes):** A ColorRef object ([MS-WMF] section 2.2.2.8) that specifies the background color of the source bitmap.

**UsageSrc (4 bytes):** An unsigned integer that specifies how to interpret values in the [**color table**](#gt_4737483c-820c-41e2-a7be-0ffed2eb4418) in the source bitmap header. This value is in the [DIBColors](#Section_a5e722e3891a4a67be1aed5a48a7fda1) enumeration (section 2.1.9).

**offBmiSrc (4 bytes):** An unsigned integer that specifies the offset in bytes, from the start of this record to the source bitmap header in the **BitmapBuffer** field.

**cbBmiSrc (4 bytes):** An unsigned integer that specifies the size in bytes, of the source bitmap header.

**offBitsSrc (4 bytes):** An unsigned integer that specifies the offset in bytes, from the start of this record to the source bitmap bits in the **BitmapBuffer** field.

**cbBitsSrc (4 bytes):** An unsigned integer that specifies the size in bytes, of the source bitmap bits.

**BitmapBuffer (variable):** A buffer containing the source bitmap, which is not required to be contiguous with the fixed portion of the EMR\_BITBLT record. Thus, fields in this buffer that are labeled "UndefinedSpace" are optional and MUST be ignored.

If the raster operation specified by **BitBltRasterOperation** does not require a source bitmap, the source bitmap can be omitted.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| UndefinedSpace1 (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BmiSrc (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| UndefinedSpace2 (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BitsSrc (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**BmiSrc (variable):** The source bitmap header.

**BitsSrc (variable):** The source bitmap bits.

See section 2.3.1 for more bitmap record types.

#### EMR\_MASKBLT Record

The EMR\_MASKBLT record specifies a block transfer of pixels from a source [**bitmap**](#gt_26a17e58-eeba-4210-81b9-8bbadf7ee952) to a destination rectangle, optionally in combination with a brush pattern and with the application of a color mask bitmap, according to specified foreground and background [**raster operations**](#gt_772681cd-b9a5-440c-8021-b6219c70f6fc).

Fields not specified in this section are specified in section [2.3.1](#Section_2a57d4bdf3dc45df95fd447382b5262a).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| xDest | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| yDest | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cxDest | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cyDest | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ROP4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| xSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ySrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| XformSrc (24 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BkColorSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| UsageSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offBmiSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbBmiSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offBitsSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbBitsSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| xMask | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| yMask | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| UsageMask | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offBmiMask | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbBmiMask | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offBitsMask | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbBitsMask | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BitmapBuffer (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_MASKBLT**. This value is 0x0000004E.

**Bounds (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19) that specifies the destination bounding rectangle in logical coordinates. If the intersection of this rectangle with the current clipping [**regions**](#gt_6f13fe5f-8747-4ae0-9375-814bf0528197) (section [3.1.1.2.1](#Section_939400bc0ecd4a28aa8c60c7e4cc9877)) in the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45) (section [3.1](#Section_50ae16a09d1748018bf1e28e9533a889)) is empty, this record has no effect.

**xDest (4 bytes):** A signed integer that specifies the logical x-coordinate of the upper-left corner of the destination rectangle.

**yDest (4 bytes):** A signed integer that specifies the logical y-coordinate of the upper-left corner of the destination rectangle.

**cxDest (4 bytes):** A signed integer that specifies the logical width of the destination rectangle.

**cyDest (4 bytes):** A signed integer that specifies the logical height of the destination rectangle.

**ROP4 (4 bytes):** A quaternary raster operation, which specifies ternary raster operations for the foreground and background colors of a bitmap. These values define how the color data of the source rectangle is to be combined with the color data of the destination rectangle.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Reserved | | | | | | | | | | | | | | | | BackgroundROP3 | | | | | | | | ForegroundROP3 | | | | | | | |

**Reserved (2 bytes):** This field SHOULD be 0x0000 and MUST be ignored.[<57>](#Appendix_A_57" \o "Product behavior note 57)

**BackgroundROP3 (1 byte):** The unsigned, most-significant 8 bits of a 24-bit ternary raster operation value from the Ternary Raster Operation enumeration ([MS-WMF] section 2.1.1.31). This code defines how to combine the background color data of the source and destination bitmaps and brush pattern.

**ForegroundROP3 (1 byte):** The unsigned, most-significant 8 bits of a 24-bit ternary raster operation value from the Ternary Raster Operation enumeration. This code defines how to combine the foreground color data of the source and destination bitmaps and brush pattern.

**xSrc (4 bytes):** A signed integer that specifies the logical x-coordinate of the upper-left corner of the source rectangle.

**ySrc (4 bytes):** A signed integer that specifies the logical y-coordinate of the upper-left corner of the source rectangle.

**XformSrc (24 bytes):** An [XForm](#Section_e84107e9bc2b4a1492345d173adc1b59) object (section 2.2.28) that specifies a [**world-space**](#gt_baadb707-1343-413b-9b36-2496b06e9bd7) to [**page-space**](#gt_f530f123-858a-4af7-b626-adea0aeaa75a) [**transform**](#gt_fdca4a96-8e04-4dc0-acad-1d01e1384a5b) to apply to the source bitmap.

**BkColorSrc (4 bytes):** A ColorRef object ([MS-WMF] section 2.2.2.8) that specifies the background color of the source bitmap.

**UsageSrc (4 bytes):** An unsigned integer that specifies how to interpret values in the [**color table**](#gt_4737483c-820c-41e2-a7be-0ffed2eb4418) in the source bitmap header. This value is in the [DIBColors](#Section_a5e722e3891a4a67be1aed5a48a7fda1) enumeration (section 2.1.9).

**offBmiSrc (4 bytes):** An unsigned integer that specifies the offset in bytes, from the start of this record to the source bitmap header in the **BitmapBuffer** field.

**cbBmiSrc (4 bytes):** An unsigned integer that specifies the size in bytes of the source bitmap header.

**offBitsSrc (4 bytes):** An unsigned integer that specifies the offset in bytes, from the start of this record to the source bitmap bits in the **BitmapBuffer** field.

**cbBitsSrc (4 bytes):** An unsigned integer that specifies the size in bytes, of the source bitmap bits.

**xMask (4 bytes):** A signed integer that specifies the logical x-coordinate of the upper-left corner of the mask bitmap.

**yMask (4 bytes):** A signed integer that specifies the logical y-coordinate of the upper-left corner of the mask bitmap.

**UsageMask (4 bytes):** An unsigned integer that specifies how to interpret values in the color table in the mask bitmap header. This value is in the DIBColors enumeration.

**offBmiMask (4 bytes):** An unsigned integer that specifies the offset in bytes, from the start of this record to the mask bitmap header in the **BitmapBuffer** field.

**cbBmiMask (4 bytes):** An unsigned integer that specifies the size in bytes, of the mask bitmap header.

**offBitsMask (4 bytes):** An unsigned integer that specifies the offset in bytes, from the start of this record to the mask bitmap bits in the **BitmapBuffer** field.

**cbBitsMask (4 bytes):** An unsigned integer that specifies the size in bytes, of the mask bitmap bits.

**BitmapBuffer (variable):** A buffer containing the source and mask bitmaps, which are not required to be contiguous with the fixed portion of the EMR\_MASKBLT record or with each other. Thus, fields in this buffer that are labeled "UndefinedSpace" are optional and MUST be ignored.

**Note:** The source and mask bitmaps can be present in this buffer in any order.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| UndefinedSpace1 (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BmiSrc (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| UndefinedSpace2 (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BitsSrc (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| UndefinedSpace3 (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BmiMask (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| UndefinedSpace4 (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BitsMask (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**BmiSrc (variable):** The source bitmap header.

**BitsSrc (variable):** The source bitmap bits.

**BmiMask (variable):** The mask bitmap header.

**BitsMask (variable):** The mask bitmap bits.

The mask bitmap MUST be monochrome; that is, each pixel value MUST be zero or one. A pixel value of one in the mask indicates that the color of the corresponding pixel in the source bitmap SHOULD be copied to the destination. A value of zero in the mask indicates that the destination pixel color SHOULD NOT be changed. If the mask rectangle is smaller than the source and destination rectangles, the mask pattern MUST be replicated as necessary.

See section 2.3.1 for more bitmap record types.

#### EMR\_PLGBLT Record

The EMR\_PLGBLT record specifies a block transfer of pixels from a source [**bitmap**](#gt_26a17e58-eeba-4210-81b9-8bbadf7ee952) to a destination parallelogram, with the application of a color mask bitmap.

Fields not specified in this section are specified in section [2.3.1](#Section_2a57d4bdf3dc45df95fd447382b5262a).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| aptlDest (24 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| xSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ySrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cxSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cySrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| XformSrc (24 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BkColorSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| UsageSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offBmiSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbBmiSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offBitsSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbBitsSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| xMask | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| yMask | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| UsageMask | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offBmiMask | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbBmiMask | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offBitsMask | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbBitsMask | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BitmapBuffer (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_PLGBLT**. This value is 0x0000004F.

**Bounds (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19) that specifies the destination bounding rectangle in logical coordinates. If the intersection of this rectangle with the current clipping [**regions**](#gt_6f13fe5f-8747-4ae0-9375-814bf0528197) (section [3.1.1.2.1](#Section_939400bc0ecd4a28aa8c60c7e4cc9877)) in the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45) (section [3.1](#Section_50ae16a09d1748018bf1e28e9533a889)) is empty, this record has no effect.

**aptlDest (24 bytes):** An array of three PointL objects ([MS-WMF] section 2.2.2.15) that specifies three corners a parallelogram destination area for the block transfer.

The upper-left corner of the source rectangle is mapped to the first point in this array, the upper-right corner to the second point, and the lower-left corner to the third point. The lower-right corner of the source rectangle is mapped to the implicit fourth point in the parallelogram, which is computed from the first three points (A, B, and C) by treating them as vectors.

1. D = B + C A

**xSrc (4 bytes):** A signed integer that specifies the logical x-coordinate of the upper-left corner of the source rectangle.

**ySrc (4 bytes):** A signed integer that specifies the logical y-coordinate of the upper-left corner of the source rectangle.

**cxSrc (4 bytes):** A signed integer that specifies the logical width of the source rectangle.

**cySrc (4 bytes):** A signed integer that specifies the logical height of the source rectangle.

**XformSrc (24 bytes):** An [XForm](#Section_e84107e9bc2b4a1492345d173adc1b59) object (section 2.2.28) that specifies a [**world-space**](#gt_baadb707-1343-413b-9b36-2496b06e9bd7) to [**page-space**](#gt_f530f123-858a-4af7-b626-adea0aeaa75a) [**transform**](#gt_fdca4a96-8e04-4dc0-acad-1d01e1384a5b) to apply to the source bitmap.

**BkColorSrc (4 bytes):** A ColorRef object ([MS-WMF] section 2.2.2.8) that specifies the background color of the source bitmap.

**UsageSrc (4 bytes):** An unsigned integer that specifies how to interpret values in the [**color table**](#gt_4737483c-820c-41e2-a7be-0ffed2eb4418) in the source bitmap header. This value is in the [DIBColors](#Section_a5e722e3891a4a67be1aed5a48a7fda1) enumeration (section 2.1.9).

**offBmiSrc (4 bytes):** An unsigned integer that specifies the offset in bytes, from the start of this record to the source bitmap header in the **BitmapBuffer** field.

**cbBmiSrc (4 bytes):** An unsigned integer that specifies the size in bytes, of the source bitmap header.

**offBitsSrc (4 bytes):** An unsigned integer that specifies the offset in bytes, from the start of this record to the source bitmap bits in the **BitmapBuffer** field.

**cbBitsSrc (4 bytes):** An unsigned integer that specifies the size in bytes, of the source bitmap.

**xMask (4 bytes):** A signed integer that specifies the logical x-coordinate of the upper-left corner of the mask bitmap.

**yMask (4 bytes):** A signed integer that specifies the logical y-coordinate of the upper-left corner of the mask bitmap.

**UsageMask (4 bytes):** An unsigned integer that specifies how to interpret values in the color table in the mask bitmap header. This value is in the DIBColors enumeration.

**offBmiMask (4 bytes):** An unsigned integer that specifies the offset in bytes, from the start of this record to the header of the mask bitmap in the **BitmapBuffer** field.

**cbBmiMask (4 bytes):** An unsigned integer that specifies the size in bytes, of the mask bitmap header.

**offBitsMask (4 bytes):** An unsigned integer that specifies the offset in bytes, from the start of this record to the mask bitmap bits in the **BitmapBuffer** field.

**cbBitsMask (4 bytes):** An unsigned integer that specifies the size in bytes, of the mask bitmap bits.

**BitmapBuffer (variable):** A buffer containing the source and mask bitmaps, which are not required to be contiguous with the fixed portion of the EMR\_PLGBLT record or with each other. Thus, fields in this buffer that are labeled "UndefinedSpace" are optional and MUST be ignored.

**Note:** The source and mask bitmaps can be present in this buffer in any order.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| UndefinedSpace1 (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BmiSrc (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| UndefinedSpace2 (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BitsSrc (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| UndefinedSpace3 (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BmiMask (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| UndefinedSpace4 (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BitsMask (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**BmiSrc (variable):** The source bitmap header.

**BitsSrc (variable):** The source bitmap bits.

**BmiMask (variable):** The mask bitmap header.

**BitsMask (variable):** The mask bitmap bits.

The mask bitmap MUST be monochrome; that is, each pixel value MUST be zero or one. A pixel value of one in the mask indicates that the color of the corresponding pixel in the source bitmap SHOULD be copied to the destination. A value of zero in the mask indicates that the destination pixel color SHOULD NOT be changed. If the mask rectangle is smaller than the source and destination rectangles, the mask pattern MUST be replicated as necessary.

See section 2.3.1 for more bitmap record types.

#### EMR\_SETDIBITSTODEVICE Record

The EMR\_SETDIBITSTODEVICE record specifies a block transfer of pixels from specified scanlines of a source [**bitmap**](#gt_26a17e58-eeba-4210-81b9-8bbadf7ee952) to a destination rectangle.

Fields not specified in this section are specified in section [2.3.1](#Section_2a57d4bdf3dc45df95fd447382b5262a).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| xDest | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| yDest | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| xSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ySrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cxSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cySrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offBmiSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbBmiSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offBitsSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbBitsSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| UsageSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| iStartScan | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cScans | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BitmapBuffer (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_SETDIBITSTODEVICE**. This value is 0x00000050.

**Bounds (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19) that specifies the destination bounding rectangle in logical coordinates. If the intersection of this rectangle with the current clipping [**regions**](#gt_6f13fe5f-8747-4ae0-9375-814bf0528197) (section [3.1.1.2.1](#Section_939400bc0ecd4a28aa8c60c7e4cc9877)) in the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45) (section [3.1](#Section_50ae16a09d1748018bf1e28e9533a889)) is empty, this record has no effect.

**xDest (4 bytes):** A signed integer that specifies the logical x-coordinate of the upper-left corner of the destination rectangle.

**yDest (4 bytes):** A signed integer that specifies the logical y-coordinate of the upper-left corner of the destination rectangle.

**xSrc (4 bytes):** A signed integer that specifies the x-coordinate in pixels of the lower-left corner of the source rectangle.

**ySrc (4 bytes):** A signed integer that specifies the y-coordinate in pixels of the lower-left corner of the source rectangle.

**cxSrc (4 bytes):** A signed integer that specifies the width in pixels of the source rectangle.

**cySrc (4 bytes):** A signed integer that specifies the height in pixels of the source rectangle.

**offBmiSrc (4 bytes):** An unsigned integer that specifies the offset in bytes, from the start of this record to the source bitmap header in the **BitmapBuffer** field.

**cbBmiSrc (4 bytes):** An unsigned integer that specifies the size in bytes, of the source bitmap header.

**offBitsSrc (4 bytes):** An unsigned integer that specifies the offset in bytes, from the start of this record to the source bitmap bits in the **BitmapBuffer** field.

**cbBitsSrc (4 bytes):** An unsigned integer that specifies the size in bytes, of the source bitmap bits.

**UsageSrc (4 bytes):** An unsigned integer that specifies how to interpret values in the [**color table**](#gt_4737483c-820c-41e2-a7be-0ffed2eb4418) in the source bitmap header. This value is in the [DIBColors](#Section_a5e722e3891a4a67be1aed5a48a7fda1) enumeration (section 2.1.9).

**iStartScan (4 bytes):** An unsigned integer that specifies the first scan line in the array.

**cScans (4 bytes):** An unsigned integer that specifies the number of scan lines.

**BitmapBuffer (variable):** A buffer containing the source bitmap, which is not required to be contiguous with the fixed portion of the EMR\_SETDIBITSTODEVICE record. Thus, fields in this buffer that are labeled "UndefinedSpace" are optional and MUST be ignored.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| UndefinedSpace1 (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BmiSrc (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| UndefinedSpace2 (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BitsSrc (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**BmiSrc (variable):** The source bitmap header.

**BitsSrc (variable):** The source bitmap bits.

This record supports source images in [**JPEG**](#gt_5eda1f18-8071-4b27-ab0f-07f1fb79199d) and [**PNG**](#gt_212c87ce-40ba-4311-be2a-494f1a116604) format. The **Compression** field in the source bitmap header specifies the image format.

See section 2.3.1 for more bitmap record types.

#### EMR\_STRETCHBLT Record

The EMR\_STRETCHBLT record specifies a block transfer of pixels from a source [**bitmap**](#gt_26a17e58-eeba-4210-81b9-8bbadf7ee952) to a destination rectangle, optionally in combination with a brush pattern, according to a specified [**raster operation**](#gt_772681cd-b9a5-440c-8021-b6219c70f6fc), stretching or compressing the output to fit the dimensions of the destination, if necessary.

Fields not specified in this section are specified in section [2.3.1](#Section_2a57d4bdf3dc45df95fd447382b5262a).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| xDest | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| yDest | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cxDest | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cyDest | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BitBltRasterOperation | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| xSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ySrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| XformSrc (24 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BkColorSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| UsageSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offBmiSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbBmiSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offBitsSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbBitsSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cxSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cySrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BitmapBuffer (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_STRETCHBLT**. This value is 0x0000004D.

**Bounds (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19) that specifies the destination bounding rectangle in logical coordinates. If the intersection of this rectangle with the current clipping [**regions**](#gt_6f13fe5f-8747-4ae0-9375-814bf0528197) (section [3.1.1.2.1](#Section_939400bc0ecd4a28aa8c60c7e4cc9877)) in the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45) (section [3.1](#Section_50ae16a09d1748018bf1e28e9533a889)) is empty, this record has no effect.

**xDest (4 bytes):** A signed integer that specifies the logical x-coordinate of the upper-left corner of the destination rectangle.

**yDest (4 bytes):** A signed integer that specifies the logical y-coordinate of the upper-left corner of the destination rectangle.

**cxDest (4 bytes):** A signed integer that specifies the logical width of the destination rectangle.

**cyDest (4 bytes):** A signed integer that specifies the logical height of the destination rectangle.

**BitBltRasterOperation (4 bytes):** An unsigned integer that specifies the raster operation code. This code defines how the color data of the source rectangle is to be combined with the color data of the destination rectangle and optionally a brush pattern, to achieve the final color.

This value is in the Ternary Raster Operation enumeration ([MS-WMF] section 2.1.1.31).

**xSrc (4 bytes):** A signed integer that specifies the logical x-coordinate of the upper-left corner of the source rectangle.

**ySrc (4 bytes):** A signed integer that specifies the logical y-coordinate of the upper-left corner of the source rectangle.

**XformSrc (24 bytes):** An [XForm](#Section_e84107e9bc2b4a1492345d173adc1b59) object (section 2.2.28) that specifies a [**world-space**](#gt_baadb707-1343-413b-9b36-2496b06e9bd7) to [**page-space**](#gt_f530f123-858a-4af7-b626-adea0aeaa75a) [**transform**](#gt_fdca4a96-8e04-4dc0-acad-1d01e1384a5b) to apply to the source bitmap.

**BkColorSrc (4 bytes):** A ColorRef object ([MS-WMF] section 2.2.2.8) that specifies the background color of the source bitmap.

**UsageSrc (4 bytes):** An unsigned integer that specifies how to interpret values in the [**color table**](#gt_4737483c-820c-41e2-a7be-0ffed2eb4418) in the source bitmap header. This value is in the [DIBColors](#Section_a5e722e3891a4a67be1aed5a48a7fda1) enumeration (section 2.1.9).

**offBmiSrc (4 bytes):** An unsigned integer that specifies the offset in bytes, from the start of this record to the source bitmap header.

**cbBmiSrc (4 bytes):** An unsigned integer that specifies the size in bytes, of the source bitmap header.

**offBitsSrc (4 bytes):** An unsigned integer that specifies the offset in bytes, from the start of this record to the source bitmap bits.

**cbBitsSrc (4 bytes):** An unsigned integer that specifies the size in bytes, of the source bitmap bits.

**cxSrc (4 bytes):** A signed integer that specifies the logical width of the source rectangle.

**cySrc (4 bytes):** A signed integer that specifies the logical height of the source rectangle.

**BitmapBuffer (variable):** A buffer containing the source bitmap, which is not required to be contiguous with the fixed portion of the EMR\_STRETCHBLT record. Thus, fields in this buffer that are labeled "UndefinedSpace" are optional and MUST be ignored.

If the raster operation specified by **BitBltRasterOperation** does not require a source bitmap, the source bitmap can be omitted.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| UndefinedSpace1 (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BmiSrc (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| UndefinedSpace2 (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BitsSrc (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**BmiSrc (variable):** The source bitmap header.

**BitsSrc (variable):** The source bitmap bits.

See section 2.3.1 for more bitmap record types.

#### EMR\_STRETCHDIBITS Record

The EMR\_STRETCHDIBITS record specifies a block transfer of pixels from a source [**bitmap**](#gt_26a17e58-eeba-4210-81b9-8bbadf7ee952) to a destination rectangle, optionally in combination with a brush pattern, according to a specified [**raster operation**](#gt_772681cd-b9a5-440c-8021-b6219c70f6fc), stretching or compressing the output to fit the dimensions of the destination, if necessary.

Fields not specified in this section are specified in section [2.3.1](#Section_2a57d4bdf3dc45df95fd447382b5262a).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| xDest | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| yDest | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| xSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ySrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cxSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cySrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offBmiSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbBmiSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offBitsSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbBitsSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| UsageSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BitBltRasterOperation | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cxDest | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cyDest | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BitmapBuffer (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_STRETCHDIBITS**. This value is 0x00000051.

**Bounds (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19) that specifies the destination bounding rectangle in logical coordinates. If the intersection of this rectangle with the current clipping [**regions**](#gt_6f13fe5f-8747-4ae0-9375-814bf0528197) (section [3.1.1.2.1](#Section_939400bc0ecd4a28aa8c60c7e4cc9877)) in the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45) (section [3.1](#Section_50ae16a09d1748018bf1e28e9533a889)) is empty, this record has no effect.

**xDest (4 bytes):** A signed integer that specifies the logical x-coordinate of the upper-left corner of the destination rectangle.

**yDest (4 bytes):** A signed integer that specifies the logical y-coordinate of the upper-left corner of the destination rectangle.

**xSrc (4 bytes):** A signed integer that specifies the x-coordinate in pixels of the upper-left corner of the source rectangle.

**ySrc (4 bytes):** A signed integer that specifies the y-coordinate in pixels of the upper-left corner of the source rectangle.

**cxSrc (4 bytes):** A signed integer that specifies the width in pixels of the source rectangle.

**cySrc (4 bytes):** A signed integer that specifies the height in pixels of the source rectangle.

**offBmiSrc (4 bytes):** An unsigned integer that specifies the offset in bytes from the start of this record to the source bitmap header.

**cbBmiSrc (4 bytes):** An unsigned integer that specifies the size in bytes, of the source bitmap header.

**offBitsSrc (4 bytes):** An unsigned integer that specifies the offset in bytes, from the start of this record to the source bitmap bits.

**cbBitsSrc (4 bytes):** An unsigned integer that specifies the size in bytes, of the source bitmap bits.

**UsageSrc (4 bytes):** An unsigned integer that specifies how to interpret values in the [**color table**](#gt_4737483c-820c-41e2-a7be-0ffed2eb4418) in the source bitmap header. This value is in the [DIBColors](#Section_a5e722e3891a4a67be1aed5a48a7fda1) enumeration (section 2.1.9).

**BitBltRasterOperation (4 bytes):** An unsigned integer that specifies a raster operation code. These codes define how the color data of the source rectangle is to be combined with the color data of the destination rectangle and optionally a brush pattern, to achieve the final color.

This value is in the Ternary Raster Operation enumeration ([MS-WMF] section 2.1.1.31).

**cxDest (4 bytes):** A signed integer that specifies the logical width of the destination rectangle.

**cyDest (4 bytes):** A signed integer that specifies the logical height of the destination rectangle.

**BitmapBuffer (variable):** A buffer containing the source bitmap, which is not required to be contiguous with the fixed portion of the EMR\_STRETCHDIBITS record. Thus, fields in this buffer that are labeled "UndefinedSpace" are optional and MUST be ignored.

If the raster operation specified by **BitBltRasterOperation** does not require a source bitmap, the source bitmap can be omitted.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| UndefinedSpace1 (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BmiSrc (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| UndefinedSpace2 (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BitsSrc (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**BmiSrc (variable):** The source bitmap header.

**BitsSrc (variable):** The source bitmap bits.

This record supports source images in [**JPEG**](#gt_5eda1f18-8071-4b27-ab0f-07f1fb79199d) and [**PNG**](#gt_212c87ce-40ba-4311-be2a-494f1a116604) formats. The **Compression** field in the source bitmap header specifies the image format.

If the signs of the source and destination height and width fields differ, this record specifies a mirror-image copy of the source bitmap to the destination. That is, if **cxSrc** and **cxDest** have different signs, a mirror image of the source bitmap along the x-axis is specified. If **cySrc** and **cyDest** have different signs, a mirror image of the source bitmap along the y-axis is specified.

See section 2.3.1 for more bitmap record types.

#### EMR\_TRANSPARENTBLT Record

The EMR\_TRANSPARENTBLT record specifies a block transfer of pixels from a source [**bitmap**](#gt_26a17e58-eeba-4210-81b9-8bbadf7ee952) to a destination rectangle, treating a specified color as transparent, stretching or compressing the output to fit the dimensions of the destination, if necessary.[<58>](#Appendix_A_58" \o "Product behavior note 58)

Fields not specified in this section are specified in section [2.3.1](#Section_2a57d4bdf3dc45df95fd447382b5262a).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| xDest | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| yDest | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cxDest | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cyDest | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| TransparentColor | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| xSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ySrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| XformSrc (24 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BkColorSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| UsageSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offBmiSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbBmiSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offBitsSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbBitsSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cxSrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cySrc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BitmapBuffer (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_TRANSPARENTBLT**. This value is 0x00000074.

**Bounds (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19) that specifies the destination bounding rectangle in logical coordinates. If the intersection of this rectangle with the current clipping [**regions**](#gt_6f13fe5f-8747-4ae0-9375-814bf0528197) (section [3.1.1.2.1](#Section_939400bc0ecd4a28aa8c60c7e4cc9877)) in the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45) (section [3.1](#Section_50ae16a09d1748018bf1e28e9533a889)) is empty, this record has no effect.

**xDest (4 bytes):** A signed integer that specifies the logical x-coordinate of the upper-left corner of the destination rectangle.

**yDest (4 bytes):** A signed integer that specifies the logical y-coordinate of the upper-left corner of the destination rectangle.

**cxDest (4 bytes):** A signed integer that specifies the logical width of the destination rectangle.

**cyDest (4 bytes):** A signed integer that specifies the logical height of the destination rectangle.

**TransparentColor (4 bytes):** A ColorRef object ([MS-WMF] section 2.2.2.8) that specifies the color in the source bitmap to be treated as transparent.

**xSrc (4 bytes):** A signed integer that specifies the logical x-coordinate of the upper-left corner of the source rectangle.

**ySrc (4 bytes):** A signed integer that specifies the logical y-coordinate of the upper-left corner of the source rectangle.

**XformSrc (24 bytes):** An [XForm](#Section_e84107e9bc2b4a1492345d173adc1b59) object (section 2.2.28) that specifies a [**world-space**](#gt_baadb707-1343-413b-9b36-2496b06e9bd7) to [**page-space**](#gt_f530f123-858a-4af7-b626-adea0aeaa75a) [**transform**](#gt_fdca4a96-8e04-4dc0-acad-1d01e1384a5b) to apply to the source bitmap.

**BkColorSrc (4 bytes):** A ColorRef object that specifies the background color of the source bitmap.

**UsageSrc (4 bytes):** An unsigned integer that specifies how to interpret values in the [**color table**](#gt_4737483c-820c-41e2-a7be-0ffed2eb4418) in the source bitmap header. This value is in the [DIBColors](#Section_a5e722e3891a4a67be1aed5a48a7fda1) enumeration (section 2.1.9).

**offBmiSrc (4 bytes):** An unsigned integer that specifies the offset in bytes, from the start of this record to the source bitmap header.

**cbBmiSrc (4 bytes):** An unsigned integer that specifies the size in bytes, of the source bitmap header.

**offBitsSrc (4 bytes):** An unsigned integer that specifies the offset in bytes, from the start of this record to the source bitmap bits.

**cbBitsSrc (4 bytes):** An unsigned integer that specifies the size in bytes, of the source bitmap bits.

**cxSrc (4 bytes):** A signed integer that specifies the logical width of the source rectangle.

**cySrc (4 bytes):** A signed integer that specifies the logical height of the source rectangle.

**BitmapBuffer (variable):** A buffer containing the source bitmap, which is not required to be contiguous with the fixed portion of the EMR\_TRANSPARENTBLT record. Thus, fields in this buffer that are labeled "UndefinedSpace" are optional and MUST be ignored.

If the source bitmap color format is 32 bits-per-pixel, only the [**alpha transparency**](#gt_134d5688-b046-4a5b-a74e-c0c11fc04eb3) value in each pixel SHOULD be copied to the destination.[<59>](#Appendix_A_59" \o "Product behavior note 59)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| UndefinedSpace1 (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BmiSrc (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| UndefinedSpace2 (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BitsSrc (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**BmiSrc (variable):** The source bitmap header.

**BitsSrc (variable):** The source bitmap bits.

See section 2.3.1 for more bitmap record types.

### Clipping Record Types

The **Clipping** record types define and manage clipping regions. The clipping [**regions**](#gt_6f13fe5f-8747-4ae0-9375-814bf0528197) used by clipping record types are part of the **Regions** state element (section [3.1.1.2.1](#Section_939400bc0ecd4a28aa8c60c7e4cc9877)) in the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45) (section [3.1](#Section_50ae16a09d1748018bf1e28e9533a889)).

**Note:** The EMR\_SETMETARGN record does not specify parameters.

The following are the clipping record types.

| Name | Section | Description |
| --- | --- | --- |
| EMR\_EXCLUDECLIPRECT | [2.3.2.1](#Section_a521411eb8774199abb6b4514b3574f8) | Specifies a clipping region that consists of the current clipping region minus the specified rectangle. |
| EMR\_EXTSELECTCLIPRGN | [2.3.2.2](#Section_c6b9f4e627f64a4da383c2daf5da11d9) | Combines the specified region with the current clipping region, using the specified mode. |
| EMR\_INTERSECTCLIPRECT | [2.3.2.3](#Section_13cd0c98d4e94ca7a79d58055bf45c79) | Specifies a clipping region from the intersection of the current clipping region and the specified rectangle. |
| EMR\_OFFSETCLIPRGN | [2.3.2.4](#Section_8bf2b60d3b194bd1b2d7c89b027ad808) | Specifies a clipping region as the current clipping region moved by a specified offset. |
| EMR\_SELECTCLIPPATH | [2.3.2.5](#Section_4a26bcf866074a098ec3a8768eadc8e8) | Specifies a clipping region as the current clipping region combined with the current [**path bracket**](#gt_b89ea7da-c0bf-4c45-9f3e-fc430d7ab601), using the specified mode. |
| EMR\_SETMETARGN | 2.3.2 | If the current metaregion is null, it is set to the current clipping region. Otherwise, the current metaregion is intersected with the current clipping region, and the result is the new metaregion.  After the operation, the current clipping region is set to null.  During playback, drawing occurs only within the intersection of the metaregion and clipping region.  This [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) record specifies no parameters. |

The generic structure of clipping records is specified as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ClippingRecordBuffer (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that defines the type of record. The clipping record types are listed in the following table. See the preceding table for descriptions of these record types.

| Name | Value |
| --- | --- |
| EMR\_OFFSETCLIPRGN | 0x0000001A |
| EMR\_SETMETARGN | 0x0000001C |
| EMR\_EXCLUDECLIPRECT | 0x0000001D |
| EMR\_INTERSECTCLIPRECT | 0x0000001E |
| EMR\_SELECTCLIPPATH | 0x00000043 |
| EMR\_EXTSELECTCLIPRGN | 0x0000004B |

**Size (4 bytes):** An unsigned integer that specifies the size in bytes of this record in the [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7). This value MUST be a multiple of 4 bytes.

**ClippingRecordBuffer (variable, optional):** An array of bytes that contains the data for the clipping record. The size of this field is a multiple of 4 bytes.

The EMR\_SETMETARGN record does not contain this field.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| ClippingRecordParm (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| AlignmentPadding (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**ClippingRecordParm (variable):** An array of bytes that contains the parameters for the clipping record.

**AlignmentPadding (variable, optional):** An array of up to 3 bytes that pads the record so that its total size is a multiple of 4 bytes. This field MUST be ignored.

See section [2.3](#Section_e0137630f3ad492cbde9e68866e255ba) for more EMF record types.

#### EMR\_EXCLUDECLIPRECT Record

The **EMR\_EXCLUDECLIPRECT** record excludes the specified rectangle from the current clipping [**region**](#gt_6f13fe5f-8747-4ae0-9375-814bf0528197).

Fields not specified in this section are specified in section [2.3.2](#Section_0ca0d18e324e452f9a4126e1a82e3e03).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Clip | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as EMR\_EXCLUDECLIPRECT. This value is 0x0000001D.

**Clip (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19) that specifies a rectangle in logical units.

The result of the intersection is saved as the new current clipping region. The lower and right edges of the specified rectangle MUST NOT be excluded from clipping.

See section 2.3.2 for more clipping record types.

#### EMR\_EXTSELECTCLIPRGN Record

The EMR\_EXTSELECTCLIPRGN record combines the specified [**region**](#gt_6f13fe5f-8747-4ae0-9375-814bf0528197) with the current clipping region using the specified mode.

Fields not specified in this section are specified in section [2.3.2](#Section_0ca0d18e324e452f9a4126e1a82e3e03).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| RgnDataSize | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| RegionMode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| RgnData (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as EMR\_EXTSELECTCLIPRGN. This value is 0x0000004B.

**RgnDataSize (4 bytes):** An unsigned integer that specifies the size of the **RgnData** field in bytes.

**RegionMode (4 bytes):** An unsigned integer that specifies the way to use the region. This value is in the RegionMode (section [2.1.29](#Section_b7f99f50dd2f45289624f74140368019)) enumeration.

**RgnData (variable):** An array of bytes that specifies a **RegionData** object (section [2.2.24](#Section_e66601f29b5c46198476ddb7b087551b)) in logical units. If **RegionMode** is **RGN\_COPY**, this data can be omitted and the clipping region SHOULD be set to the default clipping region.

See section 2.3.2 for more clipping record types.

#### EMR\_INTERSECTCLIPRECT Record

The EMR\_INTERSECTCLIPRECT record specifies a new clipping [**region**](#gt_6f13fe5f-8747-4ae0-9375-814bf0528197) from the intersection of the current clipping region and the specified rectangle.

Fields not specified in this section are specified in section [2.3.2](#Section_0ca0d18e324e452f9a4126e1a82e3e03).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Clip | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_INTERSECTCLIPRECT**. This value is 0x0000001E.

**Clip (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19) that specifies the rectangle in logical units.

The lower and right edges of the specified rectangle are excluded from the clipping region.

See section 2.3.2 for more clipping record types.

#### EMR\_OFFSETCLIPRGN Record

The EMR\_OFFSETCLIPRGN record moves the current clipping [**region**](#gt_6f13fe5f-8747-4ae0-9375-814bf0528197) in the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45) by the specified offsets.

Fields not specified in this section are specified in section [2.3.2](#Section_0ca0d18e324e452f9a4126e1a82e3e03).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Offset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as EMR\_OFFSETCLIPRGN. This value is 0x0000001A.

**Offset (8 bytes):** A PointL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.15) that specifies the horizontal and vertical offsets in logical units.

See section 2.3.2 for more clipping record types.

#### EMR\_SELECTCLIPPATH Record

The **EMR\_SELECTCLIPPATH** record sets the current clipping [**region**](#gt_6f13fe5f-8747-4ae0-9375-814bf0528197) in the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45) to the current clipping region combined with current [**path bracket**](#gt_b89ea7da-c0bf-4c45-9f3e-fc430d7ab601).

Fields not specified in this section are specified in section [2.3.2](#Section_0ca0d18e324e452f9a4126e1a82e3e03).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| RegionMode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as EMR\_SELECTCLIPPATH. This value is 0x00000043.

**RegionMode (4 bytes):** An unsigned integer that specifies how to combine the current clipping region with the current path bracket. This value is in the RegionMode enumeration (section [2.1.29](#Section_b7f99f50dd2f45289624f74140368019)).

See section 2.3.2 for more clipping record types.

### Comment Record Types

The **Comment** record types define formats for specifying arbitrary private data, embedding records in other [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7) formats, and adding new or special-purpose commands.

The following are [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) comment record types.

| Name | Section | Description |
| --- | --- | --- |
| EMR\_COMMENT | [2.3.3.1](#Section_e081b202429d4c34b21ca0ad501858a6) | Contains arbitrary private data. |
| EMR\_COMMENT\_EMFPLUS | [2.3.3.2](#Section_929b78e1b84844a59fac327cae5c2ae5) | Contains embedded [**EMF+**](#gt_41a5b58b-b050-4001-85ec-d6ed56379ee2) records ([[MS-EMFPLUS]](%5bMS-EMFPLUS%5d.pdf#Section_5f92c78964f246b59ed415a9bb0946c6) section 2.3). |
| EMR\_COMMENT\_EMFSPOOL | [2.3.3.3](#Section_a1a907e3cd444475a8cfdfad513a4243) | Contains embedded [**EMFSPOOL**](#gt_3bd355a2-4faa-40ab-bc79-0d158eba43b6) records ([[MS-EMFSPOOL]](%5bMS-EMFSPOOL%5d.pdf#Section_3d8cd6cc528742e8925f4a53afd04534) section 2.2). |
| EMR\_COMMENT\_PUBLIC | [2.3.3.4](#Section_39c91f22e4cc4e82a6050137caa8457c) | Specifies extensions to EMF processing. |

The generic structure of comment records is specified as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DataSize | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CommentRecordBuffer (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer from the RecordType enumeration (section [2.1.1](#Section_1eec80ba799b4784a9ac91597d590ae1)) that identifies this record as a comment record. This value is 0x00000046.

**Size (4 bytes):** An unsigned integer that specifies the size in bytes of this record in the metafile. This value MUST be a multiple of 4 bytes.

**DataSize (4 bytes):** An unsigned integer that specifies the size in bytes, of the **CommentIdentifier** and **CommentRecordParm** fields in the **RecordBuffer** field that follows. It MUST NOT include the size of itself or the size of the **AlignmentPadding** field, if present.

**CommentRecordBuffer (variable):** An array of bytes that contains the remainder of the comment record.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| CommentIdentifier (optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CommentRecordParm (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| AlignmentPadding (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**CommentIdentifier (4 bytes, optional):** An unsigned integer that identifies the type of comment record. See the preceding table for descriptions of these record types.

Valid comment identifier values are listed in the following table. If this field contains any other value, the comment record is processed as an EMR\_COMMENT record.

| Name | Value |
| --- | --- |
| EMR\_COMMENT\_EMFSPOOL | 0x00000000 |
| EMR\_COMMENT\_EMFPLUS | 0x2B464D45 |
| EMR\_COMMENT\_PUBLIC | 0x43494447 |

**CommentRecordParm (variable):** An array of bytes that contains the parameters for the comment record.

**AlignmentPadding (variable, optional):** An array of up to 3 bytes that pads the record so that its total size is a multiple of 4 bytes. This field MUST be ignored.

See section [2.3](#Section_e0137630f3ad492cbde9e68866e255ba) for more EMF record types.

#### EMR\_COMMENT Record

The **EMR\_COMMENT** record contains arbitrary private data.

Fields not specified in this section are specified in section [2.3.3](#Section_05940d07e1124146ac0588fc6a1f70b9).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DataSize | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PrivateData (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**PrivateData (variable, optional):** An array of bytes that specifies the private data. The first 32-bit field of this data MUST NOT be one of the predefined comment identifier values specified in section 2.3.3.

Private data is unknown to [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36); it is meaningful only to applications that know the format of the data and how to use it. EMR\_COMMENT private data records MAY[<60>](#Appendix_A_60" \o "Product behavior note 60) be ignored.

See section 2.3.3 for more comment record types.

#### EMR\_COMMENT\_EMFPLUS Record

The **EMR\_COMMENT\_EMFPLUS** record contains embedded [**EMF+**](#gt_41a5b58b-b050-4001-85ec-d6ed56379ee2) records ([[MS-EMFPLUS]](%5bMS-EMFPLUS%5d.pdf#Section_5f92c78964f246b59ed415a9bb0946c6) section 2.3).

Fields not specified in this section are specified in section [2.3.3](#Section_05940d07e1124146ac0588fc6a1f70b9).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DataSize | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CommentIdentifier | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EMFPlusRecords (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**CommentIdentifier (4 bytes):** An unsigned integer that identifies this comment record as containing EMF+ records. The value 0x2B464D45, which is the [**ASCII**](#gt_79fa85ca-ac61-467c-b819-e97dc1a7a599) string "+FME", identifies this as an EMR\_COMMENT\_EMFPLUS record.

**EMFPlusRecords (variable):** An array of bytes that contains one or more EMF+ records.

See section 2.3.3 for more comment record types.

#### EMR\_COMMENT\_EMFSPOOL Record

The **EMR\_COMMENT\_EMFSPOOL** record contains embedded [**EMFSPOOL**](#gt_3bd355a2-4faa-40ab-bc79-0d158eba43b6) records ([[MS-EMFSPOOL]](%5bMS-EMFSPOOL%5d.pdf#Section_3d8cd6cc528742e8925f4a53afd04534) section 2.2).

Fields not specified in this section are specified in section [2.3.3](#Section_05940d07e1124146ac0588fc6a1f70b9).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DataSize | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CommentIdentifier | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EMFSpoolRecordIdentifier | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EMFSpoolRecords (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**CommentIdentifier (4 bytes):** An unsigned integer that identifies this comment record as containing EMFSPOOL records. The value 0x00000000 identifies this as an EMR\_COMMENT\_EMFSPOOL record.

**EMFSpoolRecordIdentifier (4 bytes):** An unsigned integer that identifies the type of EMR\_COMMENT\_EMFSPOOL record. The value 0x544F4E46, which is the [**ASCII**](#gt_79fa85ca-ac61-467c-b819-e97dc1a7a599) string "TONE", identifies this as an EMFSPOOL font definition record ([MS-EMFSPOOL] section 2.2.3.3).

**EMFSpoolRecords (variable):** An array of bytes that contain one or more font definition records.

See section 2.3.3 for more comment record types.

#### EMR\_COMMENT\_PUBLIC Record Types

The **EMR\_COMMENT\_PUBLIC** record types specify extensions to [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) processing.

Following are the EMF public comment record types that have been defined.

| Name | Section | Description |
| --- | --- | --- |
| EMR\_COMMENT\_BEGINGROUP | [2.3.3.4.1](#Section_9f0561a0adfe4a97a45299664787c1d2) | Specifies the beginning of a group of drawing records. |
| EMR\_COMMENT\_ENDGROUP | [2.3.3.4.2](#Section_11829e5dd2594179b66bbc2c7ab6b985) | Specifies the end of a group of drawing records. |
| EMR\_COMMENT\_MULTIFORMATS | [2.3.3.4.3](#Section_4bc4608eca82426dac0e5ded2ad3fc4e) | Specifies an image in multiple graphics formats. |
| EMR\_COMMENT\_WINDOW\_METAFILE | [2.3.3.4.4](#Section_0dbabbd2c61d4534b526b9431c494da8) | Specifies an image in an embedded [**WMF**](#gt_48849cf6-d55c-47e5-b041-13b65854de2b) [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7). |

The generic structure of EMR\_COMMENT\_PUBLIC records is specified as follows.

Fields not specified in this section are specified in section [2.3.3](#Section_05940d07e1124146ac0588fc6a1f70b9).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DataSize | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CommentIdentifier | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PublicCommentIdentifier | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PublicCommentRecordBuffer (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**CommentIdentifier (4 bytes):** An unsigned integer that identifies this comment record as specifying public data. The value 0x43494447, which is the [**ASCII**](#gt_79fa85ca-ac61-467c-b819-e97dc1a7a599) string "CIDG", identifies this as an EMR\_COMMENT\_PUBLIC record.

**PublicCommentIdentifier (4 bytes):** An unsigned integer that identifies the type of public comment record. This SHOULD be one of the values listed in the preceding table, which are specified in the EmrComment enumeration (section [2.1.10](#Section_cd57866bd2ef49f09dd1d928ddc02885)), unless additional public comment record types have been implemented on the [**print server**](#gt_59fb3ddc-63cf-45df-8a90-46a6af9e00cb).

**PublicCommentRecordBuffer (variable, optional):** An array of bytes that contains the remainder of the public comment record.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| PublicCommentRecordParm (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| AlignmentPadding (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**PublicCommentRecordParm (variable):** An array of bytes that contains the parameters for the public comment record.

**AlignmentPadding (variable, optional):** An array of up to 3 bytes that pads the record so that its total size is a multiple of 4 bytes. This field MUST be ignored.

See section 2.3.3 for more comment record types.

##### EMR\_COMMENT\_BEGINGROUP Record

The **EMR\_COMMENT\_BEGINGROUP** record specifies the beginning of a group of drawing records.

Fields not specified in this section are specified in section [2.3.3](#Section_05940d07e1124146ac0588fc6a1f70b9) or [2.3.3.4](#Section_39c91f22e4cc4e82a6050137caa8457c).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DataSize | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CommentIdentifier | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PublicCommentIdentifier | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Rectangle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| nDescription | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Description (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**PublicCommentIdentifier (4 bytes):** An unsigned integer that identifies the type of public comment record as EMR\_COMMENT\_BEGINGROUP from the EmrComment enumeration (section [2.1.10](#Section_cd57866bd2ef49f09dd1d928ddc02885)). This value is 0x00000002.

**Rectangle (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19) that specifies the output rectangle in logical coordinates.

**nDescription (4 bytes):** The number of [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) characters in the optional description string that follows.

**Description (variable, optional):** A null-terminated Unicode string that describes this group of records.

This record MUST be followed by a corresponding EMR\_COMMENT\_ENDGROUP record (section [2.3.3.4.2](#Section_11829e5dd2594179b66bbc2c7ab6b985)). These record groups can be nested.

See section 2.3.3.4 for more public comment record types.

##### EMR\_COMMENT\_ENDGROUP Record

The **EMR\_COMMENT\_ENDGROUP** record specifies the end of a group of drawing records.

Fields not specified in this section are specified in section [2.3.3](#Section_05940d07e1124146ac0588fc6a1f70b9) or [2.3.3.4](#Section_39c91f22e4cc4e82a6050137caa8457c).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DataSize | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CommentIdentifier | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PublicCommentIdentifier | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**PublicCommentIdentifier (4 bytes):** An unsigned integer that identifies the type of public comment record as EMR\_COMMENT\_ENDGROUP from the EmrComment enumeration (section [2.1.10](#Section_cd57866bd2ef49f09dd1d928ddc02885)). This value is 0x00000003.

This record MUST be preceded by a corresponding. EMR\_COMMENT\_BEGINGROUP (section [2.3.3.4.1](#Section_9f0561a0adfe4a97a45299664787c1d2)). These records can be nested.

See section 2.3.3.4 for more public comment record types.

##### EMR\_COMMENT\_MULTIFORMATS Record

The **EMR\_COMMENT\_MULTIFORMATS** record specifies an image in multiple graphics formats.

Fields not specified in this section are specified in section [2.3.3](#Section_05940d07e1124146ac0588fc6a1f70b9) or [2.3.3.4](#Section_39c91f22e4cc4e82a6050137caa8457c).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DataSize | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CommentIdentifier | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PublicCommentIdentifier | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OutputRect | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CountFormats | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| aFormats (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| FormatData (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**PublicCommentIdentifier (4 bytes):** An unsigned integer that identifies the type of public comment record as EMR\_COMMENT\_MULTIFORMATS from the EmrComment enumeration (section [2.1.10)](#Section_cd57866bd2ef49f09dd1d928ddc02885). This value is 0x40000004.

**OutputRect (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19) that specifies the output rectangle, in logical coordinates.

**CountFormats (4 bytes):** An unsigned integer that specifies the number of graphics formats contained in this record.

**aFormats (variable):** A **CountFormats** length array of graphics formats, specified by EmrFormat objects (section [2.2.4](#Section_1bd63563ce674528bf285a8852ff0270)) in order of preference.

**FormatData (variable):** The image data for all graphics formats contained in this record.

The size of the data for each image is specified by the **DataSize** field in the corresponding EmrFormat object. Thus, the total size of this field is the sum of **DataSize** values in all EmrFormat objects.

The graphics format of the data for each image is specified by the **Signature** field in the corresponding EmrFormat object.

For example, an application can use this record type to specify an image in [**EPS**](#gt_ba716063-a4a2-4e6e-a657-7ba9fde7b93f) format using EpsData objects (section [2.2.6](#Section_a624b402f7aa4cb6b289612c9775f37e)). Subsequently, the [**PostScript**](#gt_31971c63-f974-47de-8de9-6b76d093890d) version of the image can be rendered if that graphics format is supported by the [**printer driver**](#gt_1a48eebd-e72c-494d-b8cb-84dfb7bc3b65) on the playback system.[<61>](#Appendix_A_61" \o "Product behavior note 61)

See section 2.3.3.4 for more public comment record types.

##### EMR\_COMMENT\_WINDOWS\_METAFILE Record

The **EMR\_COMMENT\_WINDOWS\_METAFILE** record specifies an image in an embedded [**WMF**](#gt_48849cf6-d55c-47e5-b041-13b65854de2b) [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7).

Fields not specified in this section are specified in section [2.3.3](#Section_05940d07e1124146ac0588fc6a1f70b9) or [2.3.3.4](#Section_39c91f22e4cc4e82a6050137caa8457c).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DataSize | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CommentIdentifier | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PublicCommentIdentifier | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Version | | | | | | | | | | | | | | | | Reserved | | | | | | | | | | | | | | | |
| Checksum | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Flags | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| WinMetafileSize | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| WinMetafile (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**PublicCommentIdentifier (4 bytes):** An unsigned integer that identifies the type of public comment record as EMR\_COMMENT\_WINDOWS\_METAFILE from the EmrComment enumeration (section [2.1.10](#Section_cd57866bd2ef49f09dd1d928ddc02885)). This value is 0x80000001.

**Version (2 bytes):** An unsigned integer that specifies the WMF metafile version in terms of support for DIBs, from the MetafileVersion enumeration ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.1.1.19).

**Reserved (2 bytes):** A value that MUST be 0x0000 and MUST be ignored.

**Checksum (4 bytes):** An unsigned integer that specifies the checksum for this record.

**Flags (4 bytes):** A value that MUST be 0x00000000 and MUST be ignored.

**WinMetafileSize (4 bytes):** An unsigned integer that specifies the size in bytes, of the **WinMetafile** field.

**WinMetafile (variable):** A buffer that contains the WMF metafile.

See section 2.3.3.4 for more public comment record types.

### Control Record Types

The **Control** record types define the start and end of an [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7) and its properties.

The following are EMF control record types.

| Name | Section | Description |
| --- | --- | --- |
| EMR\_EOF | [2.3.4.1](#Section_3f47fde00e6b40c187f3f4129af03aa1) | Indicates the end of the metafile and specifies a [**palette**](#gt_dca5cd91-8d01-424b-9722-c9e818562ee5). |
| EMR\_HEADER | [2.3.4.2](#Section_de081cd7351f4cc2830bd03fb55e89ab) | Indicates the start of the metafile and specifies properties of the device on which the metafile was created. |

The generic structure of control records is specified as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| RecordBuffer (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that defines the type of record. The control record types are listed in the following table. See the preceding table for descriptions of these record types.

| Name | Value |
| --- | --- |
| EMR\_HEADER | 0x00000001 |
| EMR\_EOF | 0x0000000E |

**Size (4 bytes):** An unsigned integer that specifies the size in bytes of this record in the metafile. This value MUST be a multiple of 4 bytes.

**RecordBuffer (variable):** An array of bytes that contains the remainder of the control record.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| ControlRecordParm (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| AlignmentPadding (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**ControlRecordParm (variable):** An array of bytes that contains the parameters for the control record.

**AlignmentPadding (variable, optional):** An array of up to 3 bytes that pads the record so that its total size is a multiple of 4 bytes. This field MUST be ignored.

See section [2.3](#Section_e0137630f3ad492cbde9e68866e255ba) for more EMF record types.

#### EMR\_EOF Record

The **EMR\_EOF** record indicates the end of the [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7) and specifies a [**palette**](#gt_dca5cd91-8d01-424b-9722-c9e818562ee5).

Fields not specified in this section are specified in section [2.3.4](#Section_2caf0f4781a44079931dfca6fa499869).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| nPalEntries | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offPalEntries | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PaletteBuffer (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SizeLast | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as EMR\_EOF. This value is 0x0000000E.

**nPalEntries (4 bytes):** An unsigned integer that specifies the number of palette entries.

**offPalEntries (4 bytes):** An unsigned integer that specifies the offset to the palette entries from the start of this record.

**PaletteBuffer (variable, optional):** An array of bytes that contains palette data, which is not required to be contiguous with the fixed-length portion of the EMR\_EOF record. Thus, fields in this buffer that are labeled "UndefinedSpace" are optional and MUST be ignored.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| UndefinedSpace1 (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PaletteEntries (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| UndefinedSpace2 (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**PaletteEntries (variable):** An array of LogPaletteEntry objects (section [2.2.18](#Section_c1f7b285be164112a4e60b2fd4c1d148)) that specifies the palette data.

**SizeLast (4 bytes):** An unsigned integer that MUST be the same as **Size** and MUST be the last field of the record and hence the metafile. LogPaletteEntry objects, if they exist, MUST precede this field.

See section 2.3.4 for more control record types.

#### EMR\_HEADER Record Types

The **EMR\_HEADER** record is the starting point of an [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7). It specifies properties of the device on which the image in the metafile was recorded; this information in the header record makes it possible for EMF metafiles to be independent of any specific output device.

The following are the EMR\_HEADER record types.

| Name | Section | Description |
| --- | --- | --- |
| EmfMetafileHeader | [2.3.4.2.1](#Section_80c0684b9a0e4bdb8d6d0e5a9e9673e7) | The original EMF header record. |
| EmfMetafileHeaderExtension1 | [2.3.4.2.2](#Section_120a68e1e1994768ba1bb6b0bd019989) | The header record defined in the first extension to EMF, which added support for [**OpenGL**](#gt_216becb9-a800-4a44-8206-2ec2231c9c67) records and an optional internal pixel format descriptor.[<62>](#Appendix_A_62" \o "Product behavior note 62) |
| EmfMetafileHeaderExtension2 | [2.3.4.2.3](#Section_3b393b9124db4daab5bac74434a1d67c) | The header record defined in the second extension to EMF, which added the capability of measuring display dimensions in micrometers.[<63>](#Appendix_A_63" \o "Product behavior note 63) |

EMF metafiles SHOULD be created with an **EmfMetafileHeaderExtension2** header record.

The generic structure of EMR\_HEADER records is specified as follows.

Fields not specified in this section are specified in section [2.3.4](#Section_2caf0f4781a44079931dfca6fa499869).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EmfHeader (80 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EmfHeaderRecordBuffer (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_HEADER**. This value is 0x00000001.

**EmfHeader (80 bytes):** A Header object (section [2.2.9](#Section_e4a35c41e8e343f9bc07a18e99bb866d)), which contains information about the content and structure of the metafile.

**EmfHeaderRecordBuffer (variable, optional):** An array of bytes that contains the remainder of the EMF header record.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| EmfHeaderRecordParm (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| AlignmentPadding (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**EmfHeaderRecordParm (variable):** An array of bytes that contains additional parameters for the EMF header record.

**AlignmentPadding (variable, optional):** An array of up to 3 bytes that pads the record so that its total size is a multiple of 4 bytes. This field MUST be ignored.

The value of the **Size** field can be used to distinguish between the different EMR\_HEADER record types listed earlier in this section. There are three possible headers:

* The EmfMetafileHeader record. The fixed-size part of this header is 88 bytes, and it contains a Header object (section 2.2.9).
* The EmfMetafileHeaderExtension1 record. The fixed-size part of this header is 100 bytes, and it contains a Header object and a HeaderExtension1 object (section [2.2.10](#Section_00cc8ab4ea2e4bb195691201af47a0c8)).
* The EmfMetafileHeaderExtension2 record. The fixed-size part of this header is 108 bytes, and it contains a Header object, a HeaderExtension1 object, and a HeaderExtension2 object (section [2.2.11](#Section_9e96e5cfe94949aebaa83fffd948e588)).

There are one or two optional, variable-length fields that are possible in each header: a description string and a pixel format field. In all three types of headers, the fixed-size part comes first, followed by the variable-length fields.

The algorithm shown in the following figure computes a non-negative integer variable called **HeaderSize** from the offsets and lengths of the variable-length data. The type of header is determined from that value.

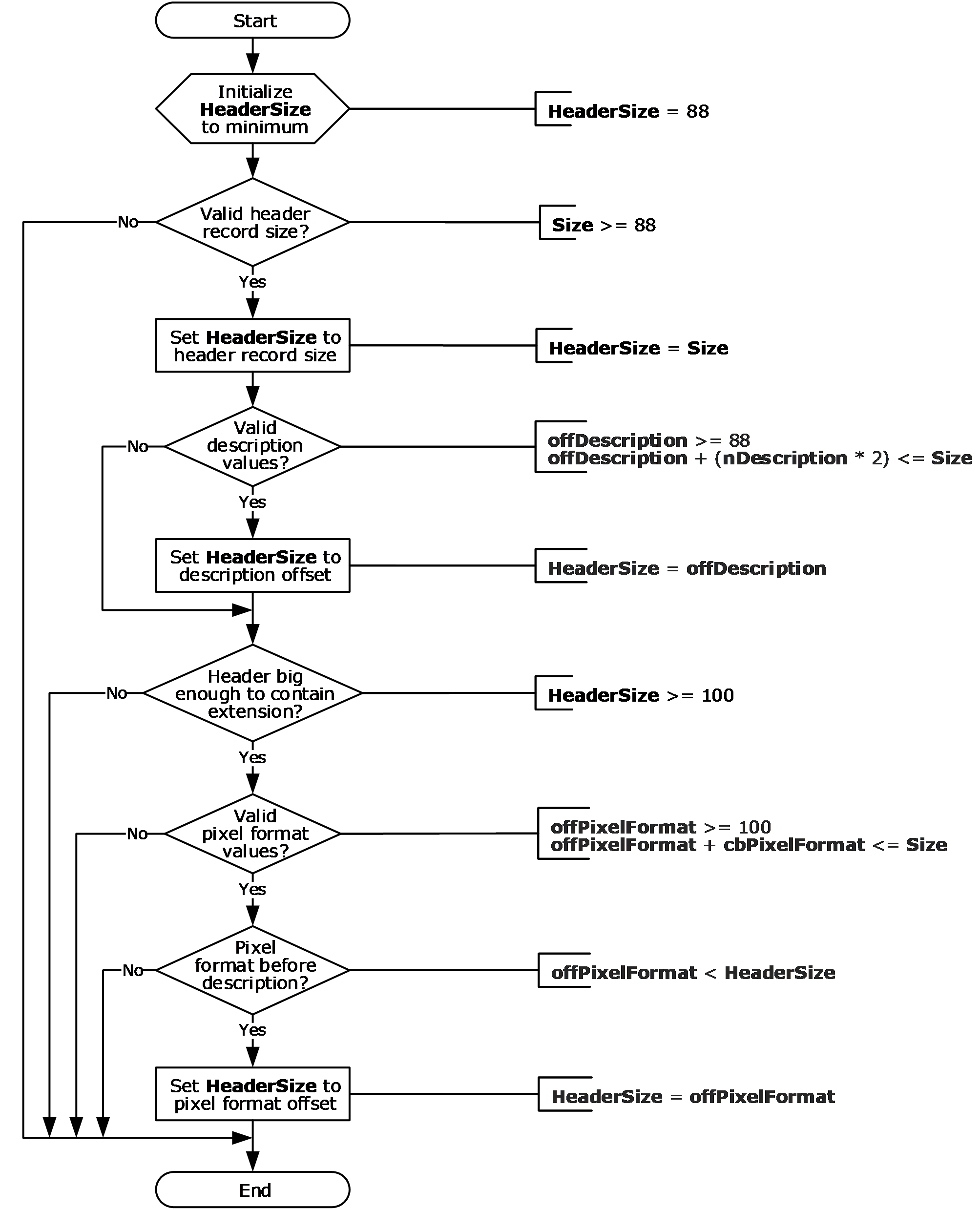


Figure 3: Header type determination algorithm

After applying the algorithm, consider the value of **HeaderSize** field:

* If **HeaderSize** >= 108, the record type is EmfMetafileHeaderExtension2.
* If **HeaderSize** >= 100, the record type is EmfMetafileHeaderExtension1.
* Otherwise, the record type is EmfMetafileHeader.

See section 2.3.4 for more control record types.

##### EmfMetafileHeader Record

The **EmfMetafileHeader** record is the header record used in the original version of [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) [**metafiles**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7).

Fields not specified in this section are specified in section [2.3.4](#Section_2caf0f4781a44079931dfca6fa499869) or [2.3.4.2](#Section_de081cd7351f4cc2830bd03fb55e89ab).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EmfHeader (80 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EmfDescriptionBuffer (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**EmfDescriptionBuffer (variable, optional):** An array of bytes that contains the EMF description string, which is not required to be contiguous with the fixed portion of this record. Thus, the undefined space field in this buffer is optional and MUST be ignored.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| UndefinedSpace (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EmfDescription (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**EmfDescription (variable):** A null-terminated [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) [**UTF16-LE**](#gt_f25550c9-f84f-4eb2-8156-14794a7e3059) string of arbitrary length and content. Its location in the record and number of characters are specified by the **offDescription** and **nDescription** fields, respectively, in **EmfHeader**. If the value of either field is zero, no description string is present.

The value of the **Size** field can be used to distinguish between the different EMR\_HEADER record types. See the flowchart in section 2.3.4.2 for details.

See section 2.3.4.2 for more header record types.

##### EmfMetafileHeaderExtension1 Record

The **EmfMetafileHeaderExtension1** record is the header record used in the first extension to [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) [**metafiles**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7). Following the **EmfHeaderExtension1** field, the remaining fields are optional and can be present in any order.

Fields not specified in this section are specified in section [2.3.4](#Section_2caf0f4781a44079931dfca6fa499869) or [2.3.4.2](#Section_de081cd7351f4cc2830bd03fb55e89ab).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EmfHeader (80 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EmfHeaderExtension1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EmfDescriptionBuffer (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EmfPixelFormatBuffer (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**EmfHeaderExtension1 (12 bytes):** A HeaderExtension1 object (section [2.2.10](#Section_00cc8ab4ea2e4bb195691201af47a0c8)), which specifies additional information about the image in the metafile.

**EmfDescriptionBuffer (variable, optional):** An array of bytes that contains the EMF description string, which is not required to be contiguous with the fixed-length part of this record. Thus, the undefined space field in this buffer is optional and MUST be ignored.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| UndefinedSpace (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EmfDescription (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**EmfDescription (variable):** A null-terminated [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) [**UTF16-LE**](#gt_f25550c9-f84f-4eb2-8156-14794a7e3059) string of arbitrary length and content. Its location in the record and number of characters are specified by the **offDescription** and **nDescription** fields, respectively, in **EmfHeader**. If the value of either field is zero, no description string is present.

**EmfPixelFormatBuffer (variable, optional):** An array of bytes that contains the EMF pixel format descriptor. It is not required to be contiguous with the **EmfHeaderExtension1** or **EmfDescription** fields. Thus, the undefined space field in this buffer is optional and MUST be ignored.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| UndefinedSpace2 (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EmfPixelFormat (40 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**EmfPixelFormat (40 bytes):** A PixelFormatDescriptor object (section [2.2.22](#Section_1db036d62da84b92b4f8e9cab8cc93b7)), which specifies the pixel format that was defined when the metafile was recorded. Its size and location in the record are specified by the **cbPixelFormat** and **offPixelFormat** fields, respectively, in **EmfHeaderExtension1**. If the value of either field is zero, no pixel format descriptor is present.

**Note:** No single structure definition can accurately represent every possible combination of optional fields. Therefore, the implementer is responsible for writing software that determines which fields are present in each metafile and for unmarshaling the contents of each field appropriately.

The value of the **Size** field can be used to distinguish between the different EMR\_HEADER record types. See the flowchart in section 2.3.4.2 for details.

##### EmfMetafileHeaderExtension2 Record

The **EmfMetafileHeaderExtension2** record is the header record used in the second extension to [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) [**metafiles**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7). Following the **EmfHeaderExtension2** field, the remaining fields are optional and can be present in any order.

Fields not specified in this section are specified in section [2.3.4](#Section_2caf0f4781a44079931dfca6fa499869) or [2.3.4.2](#Section_de081cd7351f4cc2830bd03fb55e89ab).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EmfHeader (80 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EmfHeaderExtension1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EmfHeaderExtension2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EmfDescriptionBuffer (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EmfPixelFormatBuffer (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**EmfHeaderExtension1 (12 bytes):** A HeaderExtension1 object (section [2.2.10](#Section_00cc8ab4ea2e4bb195691201af47a0c8)), which specifies additional information about the image in the metafile.

**EmfHeaderExtension2 (8 bytes):** A HeaderExtension2 object (section [2.2.11](#Section_9e96e5cfe94949aebaa83fffd948e588)), which specifies additional information about the image in the metafile.

**EmfDescriptionBuffer (variable, optional):** An array of bytes that contains the EMF description string, which is not required to be contiguous with the fixed portion of the EmfMetafileHeaderExtension2 record. Thus, the undefined space field in this buffer is optional and MUST be ignored.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| UndefinedSpace1 (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EmfDescription (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**EmfDescription (variable):** A null-terminated [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) [**UTF16-LE**](#gt_f25550c9-f84f-4eb2-8156-14794a7e3059) string of arbitrary length and content. Its location in the record and number of characters are specified by the **offDescription** and **nDescription** fields, respectively, in **EmfHeader**. If the value of either field is zero, no description string is present.

**EmfPixelFormatBuffer (variable, optional):** An array of bytes that contains the EMF pixel format descriptor, which is not required to be contiguous with the fixed portion of the EmfMetafileHeaderExtension2 record or with the EMF description string. Thus, the field in this buffer that is labeled "UndefinedSpace" is optional and MUST be ignored.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| UndefinedSpace2 (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EmfPixelFormat (40 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**EmfPixelFormat (40 bytes):** A PixelFormatDescriptor object (section [2.2.22](#Section_1db036d62da84b92b4f8e9cab8cc93b7)) that specifies the last pixel format that was defined when the metafile was recorded. Its size and location in the record are specified by the **cbPixelFormat** and **offPixelFormat** fields, respectively, in **EmfHeaderExtension1**. If the value of either field is zero, no pixel format descriptor is present.

**Note:** No single structure definition can accurately represent every possible combination of optional fields. Therefore, the implementer is responsible for writing software that determines which fields are present in each metafile, and for unmarshaling the contents of each field appropriately.

The value of the **Size** field can be used to distinguish between the different EMR\_HEADER record types. See the flowchart in section 2.3.4.2 for details.

### Drawing Record Types

The **Drawing** record types perform graphics drawing and painting functions. The clipping [**regions**](#gt_6f13fe5f-8747-4ae0-9375-814bf0528197) used by drawing records are maintained in the **Regions** state element (section [3.1.1.2.1](#Section_939400bc0ecd4a28aa8c60c7e4cc9877)) in the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45) (section [3.1](#Section_50ae16a09d1748018bf1e28e9533a889)).

The following are [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) drawing record types.

| Name | Section | Description |
| --- | --- | --- |
| EMR\_ANGLEARC | [2.3.5.1](#Section_3b04a2e29ba5477fb79c5710635d04b9) | Draws a line segment of an arc. |
| EMR\_ARC | [2.3.5.2](#Section_3e148f197e7543aa9259fda562f60315) | Draws an elliptical arc. |
| EMR\_ARCTO | [2.3.5.3](#Section_51b4a890bfd4496ebbd9dbde9ad49d1c) | Draws an elliptical arc, resetting the current drawing position to the endpoint of the arc. |
| EMR\_CHORD | [2.3.5.4](#Section_b442d2fbf92b4103aa7d44f864f719d5) | Draws a chord, which is a region bounded by the intersection of an ellipse and a line segment, called a secant. |
| EMR\_ELLIPSE | [2.3.5.5](#Section_d9462a32d18840e7bf7205614ec4ff18) | Draws an ellipse. |
| EMR\_EXTFLOODFILL | 2.3.5.1 | Draws a line segment of an arc. |
| EMR\_EXTTEXTOUTA | [2.3.5.7](#Section_6b582a713c294fc6a0f41f8a313739a1) | Draws an [**ASCII**](#gt_79fa85ca-ac61-467c-b819-e97dc1a7a599) text string using the current font and text colors. |
| EMR\_EXTTEXTOUTW | [2.3.5.8](#Section_a59a79ac328e492da34de02727af6edf) | Draws a [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) text string using the current font and text colors. |
| EMR\_FILLPATH | [2.3.5.9](#Section_7cc0afcb06934810a3cbb69c871ce473) | Closes any open figures in the current [**path bracket**](#gt_b89ea7da-c0bf-4c45-9f3e-fc430d7ab601) and fills the path's interior with the current brush. |
| EMR\_FILLRGN | [2.3.5.10](#Section_a1bb0f88bb224956b45a7005546561cc) | Fills the specified region with the specified brush. |
| EMR\_FRAMERGN | [2.3.5.11](#Section_c370fc169045467ba1257239428df6b3) | Draws a border around the specified region with the specified brush. |
| EMR\_GRADIENTFILL | [2.3.5.12](#Section_1a3849c8be6c4d30b5d3f43b4c70ca0d) | Fills the specified rectangle and triangle structures. |
| EMR\_LINETO | [2.3.5.13](#Section_9b3eccf94a554a9cb0df3c495e7b9a8c) | Draws a line from the current position up to, but not including, the specified point. This record resets the current position to that point. |
| EMR\_PAINTRGN | [2.3.5.14](#Section_68c457dc6df34609b6f9e28e5ca4d9c1) | Paints the specified region with the current brush. |
| EMR\_PIE | [2.3.5.15](#Section_c326f6b2ae7e447fb8dc02c703f054c3) | Draws a pie-shaped wedge bounded by the intersection of an ellipse and two radials. |
| EMR\_POLYBEZIER | [2.3.5.16](#Section_4e53793a95af49d4ae1f4c407eda9440) | Draws one or more [**Bezier curves**](#gt_b12730d5-b22e-4027-ba59-ff53749022a5). The cubic Bezier curves are defined with the endpoints and control points specified in this record. |
| EMR\_POLYBEZIER16 | [2.3.5.17](#Section_3329ee2617ed43718f51e3985d764d77) | Draws one or more Bezier curves with the current pen. |
| EMR\_POLYBEZIERTO | [2.3.5.18](#Section_28431e45a87441dc864d8f4f69e8e831) | Draws one or more Bezier curves based on the current position. |
| EMR\_POLYBEZIERTO16 | [2.3.5.19](#Section_d27cc96684a44715a6de45aa6cd03292) | Draws one or more Bezier curves based on the current position. |
| EMR\_POLYDRAW | [2.3.5.20](#Section_95794fafeed84d5bb6ffc6765a58166e) | Draws a set of line segments and Bezier curves. |
| EMR\_POLYDRAW16 | [2.3.5.21](#Section_57f0cfe721394199b6ad61fb61a1d4ec) | Draws a set of line segments and Bezier curves. |
| EMR\_POLYGON | [2.3.5.22](#Section_eb91678158b64e92b60668071aa65733) | Draws a polygon consisting of two or more vertexes connected by straight lines. |
| EMR\_POLYGON16 | [2.3.5.23](#Section_ac6d65ac9d39439382ebb58fa0c0f635) | Draws a polygon consisting of two or more vertexes connected by straight lines. |
| EMR\_POLYLINE | [2.3.5.24](#Section_9ce6c9bb1a1348a59aa2d95b334b5358) | Draws a series of line segments by connecting the points in the specified array. |
| EMR\_POLYLINE16 | [2.3.5.25](#Section_ca3a6590c02343548d6de3d40d7eb5b9) | Draws a series of line segments by connecting the points in the specified array. |
| EMR\_POLYLINETO | [2.3.5.26](#Section_a2d8b73883514a9c9f3aa6a8481c4c6f) | Draws one or more straight lines based upon the current position. |
| EMR\_POLYLINETO16 | [2.3.5.27](#Section_0d5a471025f943bebb083e38a4db1249) | Draws one or more straight lines based upon the current position. |
| EMR\_POLYPOLYGON | [2.3.5.28](#Section_c60ff127271142d385d4502ca2e4caef) | Paints a series of closed polygons. Each polygon is outlined with the current pen and filled with the current brush and polygon fill mode. |
| EMR\_POLYPOLYGON16 | [2.3.5.29](#Section_96ad447f25d642728d1851b395e669c6) | Paints a series of closed polygons. Each polygon is outlined with the current pen and filled with the current brush and polygon fill mode. |
| EMR\_POLYPOLYLINE | [2.3.5.30](#Section_e23f75fd3af649f38c86ef121ff61701) | Draws multiple series of connected line segments. |
| EMR\_POLYPOLYLINE16 | [2.3.5.31](#Section_a8b4cdfd50394a56b7cb9458f819f23f) | Draws multiple series of connected line segments. |
| EMR\_POLYTEXTOUTA | [2.3.5.32](#Section_d8b7ac2176b84f9aa7cc05f9d2d5627e) | Draws one or more ASCII text strings using the current font and text colors. |
| EMR\_POLYTEXTOUTW | [2.3.5.33](#Section_381015eb29a7477493ada210697f5972) | Draws one or more Unicode text strings using the current font and text colors. |
| EMR\_RECTANGLE | [2.3.5.34](#Section_3c4712380a02499290a2bfd2afd98f2a) | Draws a rectangle. The rectangle is outlined with the current pen and filled with the current brush. |
| EMR\_ROUNDRECT | [2.3.5.35](#Section_2e4f55a15f694a818329423eeedb3812) | Draws a rectangle with rounded corners. |
| EMR\_SETPIXELV | [2.3.5.36](#Section_21d44fb9b3c74ce7a4027601f34d0cda) | Defines the color of the pixel at the specified logical coordinates. |
| EMR\_SMALLTEXTOUT | [2.3.5.37](#Section_20eee81d0bd442d1a624860adfe62358) | Outputs a string. |
| EMR\_STROKEANDFILLPATH | [2.3.5.38](#Section_2794792a38b44d19adec28fc2a6273b2) | Closes any open figures in a path, draws the outline of the path with the current pen, and fills its interior with the current brush. |
| EMR\_STROKEPATH | [2.3.5.39](#Section_c8b86df6746445b0bca46ffab9174961) | Draws the specified path with the current pen. |

The generic structure of drawing records is specified as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DrawingRecordBuffer (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that defines the type of record. The drawing record types are listed in the following table. See the preceding table for descriptions of these records.

| Name | Value |
| --- | --- |
| EMR\_POLYBEZIER | 0x00000002 |
| EMR\_POLYGON | 0x00000003 |
| EMR\_POLYBEZIERTO | 0x00000005 |
| EMR\_POLYLINETO | 0x00000006 |
| EMR\_POLYPOLYLINE | 0x00000007 |
| EMR\_POLYPOLYGON | 0x00000008 |
| EMR\_SETPIXELV | 0x0000000F |
| EMR\_ANGLEARC | 0x00000029 |
| EMR\_ELLIPSE | 0x0000002A |
| EMR\_RECTANGLE | 0x0000002B |
| EMR\_ROUNDRECT | 0x0000002C |
| EMR\_ARC | 0x0000002D |
| EMR\_CHORD | 0x0000002E |
| EMR\_PIE | 0x0000002F |
| EMR\_EXTFLOODFILL | 0x00000035 |
| EMR\_LINETO | 0x00000036 |
| EMR\_ARCTO | 0x00000037 |
| EMR\_POLYDRAW | 0x00000038 |
| EMR\_FILLPATH | 0x0000003E |
| EMR\_STROKEANDFILLPATH | 0x0000003F |
| EMR\_STROKEPATH | 0x00000040 |
| EMR\_FILLRGN | 0x00000047 |
| EMR\_FRAMERGN | 0x00000048 |
| EMR\_PAINTRGN | 0x0000004A |
| EMR\_EXTTEXTOUTA | 0x00000053 |
| EMR\_EXTTEXTOUTW | 0x00000054 |
| EMR\_POLYBEZIER16 | 0x00000055 |
| EMR\_POLYGON16 | 0x00000056 |
| EMR\_POLYLINE16 | 0x00000057 |
| EMR\_POLYBEZIERTO16 | 0x00000058 |
| EMR\_POLYLINETO16 | 0x00000059 |
| EMR\_POLYPOLYLINE16 | 0x0000005A |
| EMR\_POLYPOLYGON16 | 0x0000005B |
| EMR\_POLYDRAW16 | 0x0000005C |
| EMR\_POLYTEXTOUTA | 0x00000060 |
| EMR\_POLYTEXTOUTW | 0x00000061 |
| EMR\_SMALLTEXTOUT | 0x0000006C |
| EMR\_GRADIENTFILL | 0x00000076 |

**Size (4 bytes):** An unsigned integer that specifies the size in bytes of this record in the [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7). This value MUST be a multiple of 4 bytes.

**DrawingRecordBuffer (variable):** An array of bytes that contains the remainder of the drawing record.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| DrawingRecordParm (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| AlignmentPadding (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**DrawingRecordParm (variable):** An array of bytes that contains the parameters for the drawing record.

**AlignmentPadding (variable, optional):** An array of up to 3 bytes that pads the record so that its total size is a multiple of 4 bytes. This field MUST be ignored.

See section [2.3](#Section_e0137630f3ad492cbde9e68866e255ba) for more [EMF record types](#Section_1eec80ba799b4784a9ac91597d590ae1).

#### EMR\_ANGLEARC Record

The EMR\_ANGLEARC record specifies a line segment of an arc. The line segment is drawn from the current position to the beginning of the arc. The arc is drawn along the perimeter of a circle with the given radius and center. The length of the arc is defined by the given start and sweep angles.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Center | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Radius | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| StartAngle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SweepAngle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_ANGLEARC**. This value is 0x00000029.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Center (8 bytes):** A PointL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.15), which specifies the logical coordinates of the circle's center.

**Radius (4 bytes):** An unsigned integer that specifies the circle's radius, in logical units.

**StartAngle (4 bytes):** A 32-bit float that specifies the arc's start angle, in degrees.

**SweepAngle (4 bytes):** A 32-bit float that specifies the arc's sweep angle, in degrees.

The arc is drawn by recording an imaginary circle around the specified center point with the specified radius. The starting point of the arc is determined by measuring counterclockwise from the x-axis of the circle by the number of degrees in the start angle. The ending point is similarly located by measuring counterclockwise from the starting point by the number of degrees in the sweep angle.

If the sweep angle is greater than 360 degrees, the arc is swept multiple times.

This record specifies lines by using the current pen. The figure is not filled.

See section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4) for more drawing record types.

#### EMR\_ARC Record

The EMR\_ARC record specifies an elliptical arc.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Box | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Start | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| End | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_ARC**. This value is 0x0000002D.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Box (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19), which specifies the [**inclusive-inclusive**](#gt_c706d1bc-8878-4743-9b94-e9643c5269e4) bounding rectangle in logical units.

**Start (8 bytes):** A PointL object ([MS-WMF] section 2.2.2.15), which specifies the coordinates in logical units of the ending point of the radial line defining the starting point of the arc.

**End (8 bytes):** A PointL object that specifies the coordinates in logical units of the ending point of the radial line defining the ending point of the arc.

See section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4) for more drawing record types.

#### EMR\_ARCTO Record

The EMR\_ARCTO record specifies an elliptical arc. It resets the current drawing position to the endpoint of the arc.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Box | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Start | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| End | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_ARCTO**. This value is 0x00000037.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Box (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19), which specifies the [**inclusive-inclusive**](#gt_c706d1bc-8878-4743-9b94-e9643c5269e4) bounding rectangle in logical units.

**Start (8 bytes):** A PointL object ([MS-WMF] section 2.2.2.15), which specifies the coordinates, in logical units, of the first radial ending point, in logical units.

**End (8 bytes):** A PointL object that specifies the coordinates of the second radial ending point, in logical units.

See section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4) for more drawing record types.

#### EMR\_CHORD Record

The EMR\_CHORD record specifies a chord, which is a [**region**](#gt_6f13fe5f-8747-4ae0-9375-814bf0528197) bounded by the intersection of an ellipse and a line segment, called a secant. The chord is outlined by using the current pen and filled by using the current brush.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Box | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Start | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| End | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_CHORD**. This value is 0x0000002E.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Box (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19), which specifies the [**inclusive-inclusive**](#gt_c706d1bc-8878-4743-9b94-e9643c5269e4) bounding rectangle in logical units.

**Start (8 bytes):** A PointL object ([MS-WMF] section 2.2.2.15), which specifies the coordinates, in logical units, of the endpoint of the radial defining the beginning of the chord.

**End (8 bytes):** A PointL object that specifies the logical coordinates of the endpoint of the radial defining the end of the chord.

The curve of the chord is defined by an ellipse that fits the specified bounding rectangle. The curve begins at the point where the ellipse intersects the first radial and extends counterclockwise to the point where the ellipse intersects the second radial. The chord is closed by drawing a line from the intersection of the first radial and the curve to the intersection of the second radial and the curve.

If the starting point and ending point of the curve are the same, a complete ellipse is drawn.

The current drawing position is neither used nor updated by processing this record.

See section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4) for more drawing record types.

#### EMR\_ELLIPSE Record

The EMR\_ELLIPSE record specifies an ellipse. The center of the ellipse is the center of the specified bounding rectangle. The ellipse is outlined by using the current pen and is filled by using the current brush.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Box | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_ELLIPSE**. This value is 0x0000002A.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Box (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19), which specifies the [**inclusive-inclusive**](#gt_c706d1bc-8878-4743-9b94-e9643c5269e4) bounding rectangle in logical units.

See section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4) for more drawing record types.

#### EMR\_EXTFLOODFILL Record

The EMR\_EXTFLOODFILL record fills an area of the display surface with the current brush.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Start | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Color | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| FloodFillMode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_EXTFLOODFILL**. This value is 0x00000035.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Start (8 bytes):** A PointL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.15), which specifies the coordinates, in logical units, where filling begins.

**Color (4 bytes):** A **ColorRef** object ([MS-WMF] section 2.2.2.8), which is used with the **FloodFillMode** to determine the area to fill.

**FloodFillMode (4 bytes):** An unsigned integer that specifies how to use the **Color** value to determine the area for the flood fill operation. This value is in the [FloodFill](#Section_e9e1009705f540d190ac79ccbe0b5fa9) enumeration (section 2.1.13).

See section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4) for more drawing record types.

#### EMR\_EXTTEXTOUTA Record

The **EMR\_EXTTEXTOUTA** record draws an [**ASCII**](#gt_79fa85ca-ac61-467c-b819-e97dc1a7a599) text string using the current font and text colors.

Fields not specified in this section are specified in section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| iGraphicsMode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| exScale | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| eyScale | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| aEmrText (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies the record type as EMR\_EXTTEXTOUTA from the RecordType enumeration (section [2.1.1](#Section_1eec80ba799b4784a9ac91597d590ae1)). This value is 0x00000053.

**Bounds (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19), which is not used and MUST be ignored on receipt.

**iGraphicsMode (4 bytes):** An unsigned integer that specifies the graphics mode from the GraphicsMode enumeration (section [2.1.16](#Section_fe60a5a523044b639949e11bfdbebae9)).

**exScale (4 bytes):** A FLOAT value that specifies the scale factor to apply along the X axis to convert from [**page space**](#gt_f530f123-858a-4af7-b626-adea0aeaa75a) units to .01mm units. This SHOULD be used only if the graphics mode specified by **iGraphicsMode** is GM\_COMPATIBLE.

**eyScale (4 bytes):** A FLOAT value that specifies the scale factor to apply along the Y axis to convert from page space units to .01mm units. This SHOULD be used only if the graphics mode specified by **iGraphicsMode** is GM\_COMPATIBLE.

**aEmrText (variable):** An EmrText object (section [2.2.5](#Section_dd585d0a5d7c4034963a1141af836972)) that specifies the output string in 8-bit ASCII characters, text attributes, and spacing values.

The font and text colors used for output are specified by the state of the current graphics environment (section [3.1.1.2](#Section_00de3e5e6f104be4943d8de5e2b95fe8)). A rectangle for clipping and/or opaquing can be defined in the EmrText object in the **aEmrText** field.

This record SHOULD[<64>](#Appendix_A_64" \o "Product behavior note 64) be emulated with an EMR\_EXTTEXTOUTW record (section [2.3.5.8](#Section_a59a79ac328e492da34de02727af6edf)), which requires the ASCII text string in the EmrText object to be converted to [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) [**UTF16-LE**](#gt_f25550c9-f84f-4eb2-8156-14794a7e3059) encoding.

See section 2.3.5 for more drawing record types.

#### EMR\_EXTTEXTOUTW Record

The EMR\_EXTTEXTOUTW record draws a [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) text string using the current font and text colors.

Fields not specified in this section are specified in section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| iGraphicsMode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| exScale | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| eyScale | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| wEmrText (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies the record type as **EMR\_EXTTEXTOUTW** from the [RecordType enumeration (section 2.1.1)](#Section_1eec80ba799b4784a9ac91597d590ae1). This value is 0x00000054.

**Bounds (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19). It is not used and MUST be ignored on receipt.

**iGraphicsMode (4 bytes):** An unsigned integer that specifies the current graphics mode from the [GraphicsMode enumeration (section 2.1.16)](#Section_fe60a5a523044b639949e11bfdbebae9).

**exScale (4 bytes):** A FLOAT value that specifies the scale factor to apply along the X axis to convert from [**page space**](#gt_f530f123-858a-4af7-b626-adea0aeaa75a) units to .01mm units. This is used only if the graphics mode specified by **iGraphicsMode** is **GM\_COMPATIBLE**.

**eyScale (4 bytes):** A FLOAT value that specifies the scale factor to apply along the Y axis to convert from page space units to .01mm units. This is used only if the graphics mode specified by **iGraphicsMode** is **GM\_COMPATIBLE**.

**wEmrText (variable):** An [EmrText object (section 2.2.5)](#Section_dd585d0a5d7c4034963a1141af836972) that specifies the output string in Unicode [**UTF16-LE**](#gt_f25550c9-f84f-4eb2-8156-14794a7e3059) characters, with text attributes and spacing values.

The font and text colors used for output are specified by properties in the current state of EMF metafile playback (section [3.1](#Section_50ae16a09d1748018bf1e28e9533a889)). A rectangle for clipping and/or opaquing can be defined in the EmrText object that is specified in the **aEmrText** field.

See section 2.3.5 for more drawing record types.

#### EMR\_FILLPATH Record

The EMR\_FILLPATH record closes any open figures in the current [**path bracket**](#gt_b89ea7da-c0bf-4c45-9f3e-fc430d7ab601) and fills the path's interior by using the current brush and polygon-filling mode.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_FILLPATH**. This value is 0x0000003E.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Bounds (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19), which specifies the bounding rectangle in logical units.

See section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4) for more drawing record types.

#### EMR\_FILLRGN Record

The EMR\_FILLRGN record fills the specified [**region**](#gt_6f13fe5f-8747-4ae0-9375-814bf0528197) by using the specified brush. The current clipping regions used by this record are maintained in a **Regions** state element (section [3.1.1.2.1](#Section_939400bc0ecd4a28aa8c60c7e4cc9877)) in the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45) (section [3.1](#Section_50ae16a09d1748018bf1e28e9533a889)).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| RgnDataSize | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ihBrush | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| RgnData (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_FILLRGN**. This value is 0x00000047.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Bounds (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19) that specifies the destination bounding rectangle in logical units. If the intersection of this rectangle with the current clipping region is empty, this record has no effect.

**RgnDataSize (4 bytes):** An unsigned integer that specifies the size of region data in bytes.

**ihBrush (4 bytes):** An unsigned integer that specifies the index of the brush in the EMF object table (section [3.1.1.1](#Section_e4fa4e6390964cdcb77685e2a1e4e1f4)) for filling the region.

**RgnData (variable):** A **RgnDataSize** length array of bytes that specifies the output region in a **RegionData** object (section [2.2.24](#Section_e66601f29b5c46198476ddb7b087551b)). The bounds specified by the **RegionDataHeader** field of this object MAY[<65>](#Appendix_A_65" \o "Product behavior note 65) be used as the bounding region when this record is processed.

See section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4) for more drawing record types.

#### EMR\_FRAMERGN Record

The EMR\_FRAMERGN record draws a border around the specified [**region**](#gt_6f13fe5f-8747-4ae0-9375-814bf0528197) using the specified brush. The current clipping regions used by this record are maintained in a **Regions** state element (section [3.1.1.2.1](#Section_939400bc0ecd4a28aa8c60c7e4cc9877)) in the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45) (section [3.1](#Section_50ae16a09d1748018bf1e28e9533a889)).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| RgnDataSize | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ihBrush | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Width | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Height | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| RgnData (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_FRAMERGN**. This value is 0x00000048.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Bounds (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19) that specifies the destination bounding rectangle in logical coordinates. If the intersection of this rectangle with the current clipping region is empty, this record has no effect.

**RgnDataSize (4 bytes):** An unsigned integer that specifies the size of region data in bytes.

**ihBrush (4 bytes):** An unsigned integer that specifies the index of the brush in the EMF object table index.

**Width (4 bytes):** A signed integer that specifies the width of the vertical brush stroke, in logical units.

**Height (4 bytes):** A signed integer that specifies the height of the horizontal brush stroke, in logical units.

**RgnData (variable):** A **RgnDataSize** length array of bytes that specifies the output region in a **RegionData** object (section [2.2.24](#Section_e66601f29b5c46198476ddb7b087551b)). The bounds specified by the **RegionDataHeader** field of this object MAY[<66>](#Appendix_A_66" \o "Product behavior note 66) be used as the bounding region when this record is processed.

See section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4) for more drawing record types.

#### EMR\_GRADIENTFILL Record

The **EMR\_GRADIENTFILL** record specifies filling rectangles or triangles with gradients of color.[<67>](#Appendix_A_67" \o "Product behavior note 67)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| nVer | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| nTri | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ulMode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| VertexData (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as EMR\_GRADIENTFILL. This value is 0x00000076.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Bounds (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19), which specifies the [**inclusive-inclusive**](#gt_c706d1bc-8878-4743-9b94-e9643c5269e4) bounding rectangle in logical units.

**nVer (4 bytes):** An unsigned integer that specifies the number of vertexes.

**nTri (4 bytes):** An unsigned integer that specifies the number of rectangles or triangles to fill.

**ulMode (4 bytes):** An unsigned integer that specifies the gradient fill mode. This value is in the GradientFill enumeration (section [2.1.15](#Section_699e75a028af4d6fb7aadc76e1b0001a)).

**VertexData (variable):** Objects that specify the vertexes of either rectangles or triangles and the colors that correspond to them.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| VertexObjects (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| VertexIndexes (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| VertexPadding (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**VertexObjects (variable):** An array of **nVer** [TriVertex objects (section 2.2.26)](#Section_33578509834946b68f8f107c3f70bace). Each object specifies the position and color of a vertex of either a rectangle or a triangle, depending on the value of the **ulMode** field.

**VertexIndexes (variable):** An array of **nTri** [GradientRectangle objects (section 2.2.7)](#Section_93ba94f784fc4dd58c943d476aa7588b) or [GradientTriangle objects (section 2.2.8)](#Section_9aa3346cdc29448d9dcf6a4f6b77d229), depending on the value of the **ulMode** field. Each object specifies indexes into the array of TriVertex objects in the **VertexObjects** field.

**VertexPadding (variable, optional):** An array of **nTri** times four bytes that MUST be present if the value of the **ulMode** field indicates GradientRectangle objects (section 2.2.7). If the value of the **ulMode** field indicates GradientTriangle objects (section 2.2.8), no **VertexPadding** is present. This field MUST be ignored.

An EMR\_GRADIENTFILL record that specifies that the three vertexes of a triangle SHOULD fill the figure with smooth gradients of colors.[<68>](#Appendix_A_68" \o "Product behavior note 68)

An EMR\_GRADIENTFILL record that specifies that the upper-left and lower-right vertexes of a rectangle SHOULD fill the figure with smooth gradients of color. There are two gradient fill modes in the GradientFill enumeration that can be used when drawing a rectangle. In **GRADIENT\_FILL\_RECT\_H** mode, the rectangle is filled from left to right. In **GRADIENT\_FILL\_RECT\_V** mode, the rectangle is filled from top to bottom.

An EMR\_GRADIENTFILL record MUST ignore the **Alpha** fields in the TriVertex objects. An [EMR\_ALPHABLEND record (section 2.3.1.1)](#Section_34e07d4faee64b63a4bb96996ad47669) that immediately follows the EMR\_GRADIENTFILL record can be used to apply an [**alpha transparency**](#gt_134d5688-b046-4a5b-a74e-c0c11fc04eb3) gradient to the filled area.

See section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4) for more drawing record types.

#### EMR\_LINETO Record

The EMR\_LINETO record specifies a line from the current drawing position up to, but not including, the specified point. It resets the current position to the specified point.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Point | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_LINETO**. This value is 0x00000036.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Point (8 bytes):** A PointL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.15), which specifies the coordinates of the line's endpoint.

See section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4) for more drawing record types.

#### EMR\_PAINTRGN Record

The EMR\_PAINTRGN record paints the specified [**region**](#gt_6f13fe5f-8747-4ae0-9375-814bf0528197) by using the current brush. The current clipping regions used by this record are maintained in a **Regions** state element (section [3.1.1.2.1](#Section_939400bc0ecd4a28aa8c60c7e4cc9877)) in the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45) (section [3.1](#Section_50ae16a09d1748018bf1e28e9533a889)).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| RgnDataSize | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| RgnData (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_PAINTRGN**. This value is 0x0000004A.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Bounds (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19) that specifies the destination bounding rectangle in logical coordinates. If the intersection of this rectangle with the current clipping region is empty, this record has no effect.

**RgnDataSize (4 bytes):** An unsigned integer that specifies the size of the **RgnData** field data in bytes.

**RgnData (variable):** An array of bytes that specifies the output region in a **RegionData** object (section [2.2.24](#Section_e66601f29b5c46198476ddb7b087551b)). The bounds specified by the **RegionDataHeader** field of that object MAY[<69>](#Appendix_A_69" \o "Product behavior note 69) be used as the bounding rectangle of the region when this record is processed.

See section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4) for more drawing record types.

#### EMR\_PIE Record

The EMR\_PIE record specifies a pie-shaped wedge bounded by the intersection of an ellipse and two radials. The pie is outlined by using the current pen and filled by using the current brush.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Box | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Start | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| End | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_PIE**. This value is 0x0000002F.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Box (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19), which specifies the [**inclusive-inclusive**](#gt_c706d1bc-8878-4743-9b94-e9643c5269e4) bounding rectangle in logical units.

**Start (8 bytes):** A PointL object ([MS-WMF] section 2.2.2.15), which specifies the coordinates, in logical units, of the endpoint of the first radial.

**End (8 bytes):** A PointL object that specifies the coordinates, in logical units, of the endpoint of the second radial.

The curve of the pie is defined by an ellipse that fits the specified bounding rectangle. The curve begins at the point where the ellipse intersects the first radial and extends counterclockwise to the point where the ellipse intersects the second radial.

The current drawing position is neither used nor updated by this record.

See section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4) for more drawing record types.

#### EMR\_POLYBEZIER Record

The EMR\_POLYBEZIER record specifies one or more [**Bezier curves**](#gt_b12730d5-b22e-4027-ba59-ff53749022a5).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Count | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| aPoints (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_POLYBEZIER**. This value is 0x00000002.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Bounds (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19) that specifies the bounding rectangle in logical units.

**Count (4 bytes):** An unsigned integer that specifies the number of points in the **aPoints** array. This value MUST be one more than three times the number of curves to be drawn because each Bezier curve requires two control points and an endpoint, and the initial curve requires an additional starting point.

| Line width | Device supports wideline | Maximum points allowed |
| --- | --- | --- |
| 1 | n/a | 16K |
| > 1 | yes | 16K |
| > 1 | no | 1360 |

Any extra points MUST be ignored.

**aPoints (variable):** An array of PointL objects ([MS-WMF] section 2.2.2.15) that specify the endpoints and control points of the Bezier curves in logical units.

Cubic Bezier curves are defined using the endpoints and control points specified by the **aPoints** field. The first curve is drawn from the first point to the fourth point, using the second and third points as control points. Each subsequent curve in the sequence needs exactly three more points: the ending point of the previous curve is used as the starting point, the next two points in the sequence are control points, and the third is the ending point.

The cubic Bezier curves SHOULD be drawn using the current pen.

See section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4) for more drawing record types.

#### EMR\_POLYBEZIER16 Record

The EMR\_POLYBEZIER16 record specifies one or more Bezier curves. The curves are drawn using the current pen.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Count | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| aPoints (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_POLYBEZIER16**. This value is 0x00000055.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Bounds (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19), which specifies the [**inclusive-inclusive**](#gt_c706d1bc-8878-4743-9b94-e9643c5269e4) bounding rectangle in logical units.

**Count (4 bytes):** An unsigned integer that specifies the total number of points. This value MUST be one more than three times the number of curves to be drawn because each [**Bezier curve**](#gt_b12730d5-b22e-4027-ba59-ff53749022a5) requires two control points and an endpoint, and the initial curve requires an additional starting point.

**aPoints (variable):** An array of **PointS** objects ([MS-WMF] section 2.2.2.16), which specify the points of the Bezier curves in logical units.

See section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4) for more drawing record types.

#### EMR\_POLYBEZIERTO Record

The EMR\_POLYBEZIERTO record specifies one or more [**Bezier curves**](#gt_b12730d5-b22e-4027-ba59-ff53749022a5) based upon the current drawing position.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Count | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| aPoints (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_POLYBEZIERTO**. This value is 0x00000005.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Bounds (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19) that specifies the bounding rectangle in logical units.

**Count (4 bytes):** An unsigned integer that specifies the number of points in the **aPoints** array. The first curve MUST be drawn from the current position to the third point by using the first two points as control points. For each subsequent curve, exactly three more points MUST be specified, and the ending point of the previous curve MUST be used as the starting point for the next.

| Line width | Device supports wideline | Maximum points allowed |
| --- | --- | --- |
| 1 | n/a | 16K |
| > 1 | yes | 16K |
| > 1 | no | 1360 |

Any extra points MUST be ignored.

**aPoints (variable):** An array of PointL objects ([MS-WMF] section 2.2.2.15), which specify the endpoints and control points of the Bezier curves in logical units.

The Bezier curves SHOULD be drawn using the current pen.

See section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4) for more drawing record types.

#### EMR\_POLYBEZIERTO16 Record

The EMR\_POLYBEZIERTO16 record specifies one or more Bezier curves based on the current drawing position.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Count | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| aPoints (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_POLYBEZIERTO16**. This value is 0x00000058.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Bounds (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19) that specifies the bounding rectangle in logical units.

**Count (4 bytes):** An unsigned integer that specifies the total number of points. The first curve is drawn from the current position to the third point by using the first two points as control points. For each subsequent curve, three more points MUST be specified, and the ending point of the previous curve MUST be used as the starting point for the next.

**aPoints (variable):** An array of **PointS** objects ([MS-WMF] section 2.2.2.16), which specify the points of the [**Bezier curves**](#gt_b12730d5-b22e-4027-ba59-ff53749022a5) in logical units.

See section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4) for more drawing record types.

#### EMR\_POLYDRAW Record

The EMR\_POLYDRAW record specifies a set of line segments and [**Bezier curves**](#gt_b12730d5-b22e-4027-ba59-ff53749022a5).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Count | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| aPoints (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| abTypes (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_POLYDRAW**. This value is 0x00000038.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Bounds (16 bytes):** A RectL object, specified in [[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19, which specifies the bounding rectangle in logical units.

**Count (4 bytes):** An unsigned integer that specifies the number of points in the **aPoints** field.

**aPoints (variable):** An array of PointL objects ([MS-WMF] section 2.2.2.15), which specify the points in logical units.

**abTypes (variable):** A **Count** length array of byte values that specifies how each point in the **aPoints** array is used. This value is in the Point (section [2.1.26](#Section_01b63e3f19d6437c8fc62758884bc08d)) enumeration.

See section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4) for more drawing record types.

#### EMR\_POLYDRAW16 Record

The EMR\_POLYDRAW16 record specifies a set of line segments and Bezier curves.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Count | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| aPoints (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| abTypes (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_POLYDRAW16**. This value is 0x0000005C.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Bounds (16 bytes):** A RectL object, specified in [[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19, which specifies the bounding rectangle in logical units.

**Count (4 bytes):** An unsigned integer that specifies the number of points.

**aPoints (variable):** A **Count** length array of **PointS** objects, specified in [MS-WMF] section 2.2.2.16, which specifies the array of points.

**abTypes (variable):** A **Count** length array of bytes that specifies the point types. This value is in the [Point (section 2.1.26)](#Section_01b63e3f19d6437c8fc62758884bc08d) enumeration.

See section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4) for more drawing record types.

#### EMR\_POLYGON Record

The EMR\_POLYGON record specifies a polygon consisting of two or more vertexes connected by straight lines.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Count | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| aPoints (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_POLYGON**. This value is 0x00000003.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Bounds (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19) that specifies the bounding rectangle in logical units.

**Count (4 bytes):** An unsigned integer that specifies the number of points in the **aPoints** array.

| Line width | Device supports wideline | Maximum points allowed |
| --- | --- | --- |
| 1 | n/a | 16K |
| > 1 | yes | 16K |
| > 1 | no | 1360 |

Any extra points MUST be ignored.

**aPoints (variable):** A **Count** length array of PointL objects ([MS-WMF] section 2.2.2.15) that specifies the vertexes of the polygon in logical units.

The polygon SHOULD be outlined using the current pen and filled using the current brush and polygon fill mode. The polygon SHOULD be closed automatically by drawing a line from the last vertex to the first.

See section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4) for more drawing record types.

#### EMR\_POLYGON16 Record

The EMR\_POLYGON16 record specifies a polygon consisting of two or more vertexes connected by straight lines. The polygon is outlined by using the current pen and filled by using the current brush and polygon fill mode. The polygon is closed automatically by drawing a line from the last vertex to the first.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Count | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| aPoints (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_POLYGON16**. This value is 0x00000056.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Bounds (16 bytes):** A RectL object, specified in [[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19, which specifies the bounding rectangle in logical units.

**Count (4 bytes):** An unsigned integer that specifies the total number of points.

**aPoints (variable):** A **Count** length array of **PointS** objects, specified in [MS-WMF] section 2.2.2.16, which specifies the array of points.

See section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4) for more drawing record types.

#### EMR\_POLYLINE Record

The EMR\_POLYLINE record specifies a series of line segments by connecting the points in the specified array.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Count | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| aPoints (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_POLYLINE**. This value is 0x00000004.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Bounds (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19) that specifies the bounding rectangle in logical units.

**Count (4 bytes):** An unsigned integer that specifies the number of points in the **aPoints** array.

| Line width | Device supports wideline | Maximum points allowed |
| --- | --- | --- |
| 1 | n/a | 16K |
| > 1 | yes | 16K |
| > 1 | no | 1360 |

Any extra points MUST be ignored.

**aPoints (variable):** A **Count** length array of PointL objects ([MS-WMF] section 2.2.2.15) that specifies the point data, in logical units.

The line segments SHOULD be drawn using the current pen.

See section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4) for more drawing record types.

#### EMR\_POLYLINE16 Record

The EMR\_POLYLINE16 record specifies a series of line segments by connecting the points in the specified array.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Count | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| aPoints (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_POLYLINE16**. This value is 0x00000057.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Bounds (16 bytes):** A RectL object, specified in [[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19, which specifies the bounding rectangle in logical units.

**Count (4 bytes):** An unsigned integer that specifies the total number of points.

**aPoints (variable):** A **Count** length array of **PointS** objects, specified in [MS-WMF] section 2.2.2.16, which specifies the array of points.

See section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4) for more drawing record types.

#### EMR\_POLYLINETO Record

The EMR\_POLYLINETO record specifies one or more straight lines based upon the current drawing position.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Count | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| aPoints (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_POLYLINETO**. This value is 0x00000006.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Bounds (16 bytes):** A RectL object, specified in [[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19, which specifies the bounding rectangle in logical units.

**Count (4 bytes):** An unsigned integer that specifies the number of points in the **aPoints** array.

| Line width | Device supports wideline | Maximum points allowed |
| --- | --- | --- |
| 1 | n/a | 16K |
| > 1 | yes | 16K |
| > 1 | no | 1360 |

Any extra points MUST be ignored.

**aPoints (variable):** A **Count** length array of PointL objects ([MS-WMF] section 2.2.2.15, which specifies the point data, in logical units.

A line SHOULD be drawn from the current position to the first point specified by the **aPoints** field using the current pen. Each additional line SHOULD be drawn from the ending point of the previous line to the next point specified by **aPoints**.

See section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4) for more drawing record types.

#### EMR\_POLYLINETO16 Record

The EMR\_POLYLINETO16 record specifies one or more straight lines based upon the current drawing position.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Count | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| aPoints (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_POLYLINETO16**. This value is 0x00000059.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Bounds (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19, which specifies the bounding rectangle in logical units.

**Count (4 bytes):** An unsigned integer that specifies the number of points.

**aPoints (variable):** A **Count** length array of **PointS** objects ([MS-WMF] section 2.2.2.16, which specifies the array of points.

A line is drawn from the current drawing position to the first point specified by the **aPoints** field by using the current pen. For each additional line, drawing is performed from the ending point of the previous line to the next point specified by **aPoints**.

See section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4) for more drawing record types.

#### EMR\_POLYPOLYGON Record

The EMR\_POLYPOLYGON record specifies a series of closed polygons.

Fields not specified in this section are specified in section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NumberOfPolygons | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Count | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PolygonPointCount (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| aPoints (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_POLYPOLYGON**. This value is 0x00000008.

**Bounds (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19) that specifies the bounding rectangle in logical units.

**NumberOfPolygons (4 bytes):** An unsigned integer that specifies the number of polygons.

**Count (4 bytes):** An unsigned integer that specifies the total number of points in all polygons.

| Line width | Device supports wideline | Maximum points allowed |
| --- | --- | --- |
| 1 | n/a | 16K |
| > 1 | yes | 16K |
| > 1 | no | 1360 |

Any extra points MUST be ignored. To draw a line with more points, the data SHOULD be divided into groups that have less than the maximum number of points, and an EMR\_POLYPOLYGON operation SHOULD be performed for each group of points.

**PolygonPointCount (variable):** An array of 32-bit unsigned integers that specifies the point count for each polygon.

**aPoints (variable):** An array of PointL objects ([MS-WMF] section 2.2.2.15) that specifies the points for all polygons in logical units. The number of points is specified by the **Count** field value.

Each polygon SHOULD be outlined using the current pen, and filled using the current brush and polygon fill mode that are defined in the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45). The polygons defined by this record can overlap.

See section 2.3.5 for more drawing record types.

#### EMR\_POLYPOLYGON16 Record

The EMR\_POLYPOLYGON16 record specifies a series of closed polygons. Each polygon is outlined using the current pen, and filled using the current brush and polygon fill mode. The polygons drawn by this record can overlap.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NumberOfPolygons | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Count | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PolygonPointCount (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| aPoints (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_POLYPOLYGON16**. This value is 0x0000005B.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Bounds (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19), which specifies the bounding rectangle in logical units.

**NumberOfPolygons (4 bytes):** An unsigned integer that specifies the number of polygons.

**Count (4 bytes):** An unsigned integer that specifies the total number of points in all polygons.

**PolygonPointCount (variable):** A **NumberOfPolygons** length array of 32-bit unsigned integers that specifies the point counts for each polygon.

**aPoints (variable):** A **Count** length array of **PointS** objects ([MS-WMF] section 2.2.2.16), which specifies the array of points.

See section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4) for more drawing record types.

#### EMR\_POLYPOLYLINE Record

The EMR\_POLYPOLYLINE record draws multiple series of connected line segments.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NumberOfPolylines | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Count | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| aPolylinePointCount (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| aPoints (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_POLYPOLYLINE**. This value is 0x00000007.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Bounds (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19) that specifies the bounding rectangle in logical units.

**NumberOfPolylines (4 bytes):** An unsigned integer that specifies the number of polylines, which is the number of elements in the **aPolylinePointCount** array.

**Count (4 bytes):** An unsigned integer that specifies the total number of points in all polylines, which is the number of elements in the **aPoints** array.

| Line width | Device supports wideline | Maximum points allowed |
| --- | --- | --- |
| 1 | n/a | 16K |
| > 1 | yes | 16K |
| > 1 | no | 1360 |

Any extra points MUST be ignored.

**aPolylinePointCount (variable):** A **NumberOfPolylines**-length array of 32-bit unsigned integers that specify the point counts for all polylines. Each value MUST be >= 0x00000002.

Each point count refers to a number of consecutive elements in the **aPoints** array.

**aPoints (variable):** A **Count**-length array of PointL objects ([MS-WMF] section 2.2.2.15) that specify the point data, in logical units.

The line segments SHOULD be drawn using the current pen. The figures formed by the segments SHOULD NOT filled. The current drawing position SHOULD neither be used nor updated by this record.

See section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4) for more drawing record types.

#### EMR\_POLYPOLYLINE16 Record

The EMR\_POLYPOLYLINE16 record specifies multiple series of connected line segments.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NumberOfPolylines | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Count | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PolylinePointCount (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| aPoints (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_POLYPOLYLINE16**. This value is 0x0000005A.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Bounds (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19), which specifies the bounding rectangle in logical units.

**NumberOfPolylines (4 bytes):** An unsigned integer that specifies the number of polylines.

**Count (4 bytes):** An unsigned integer that specifies the total number of points in all polylines.

**PolylinePointCount (variable):** A **NumberOfPolylines** length array of 32-bit unsigned integers that specifies the point counts for each polyline.

**aPoints (variable):** A **Count** length array of **PointS** objects ([MS-WMF] section 2.2.2.16), which specifies the array of points.

See section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4) for more drawing record types.

#### EMR\_POLYTEXTOUTA Record

The EMR\_POLYTEXTOUTA record draws one or more [**ASCII**](#gt_79fa85ca-ac61-467c-b819-e97dc1a7a599) text strings using the current font and text colors.

Fields not specified in this section are specified in section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| iGraphicsMode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| exScale | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| eyScale | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cStrings | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| aEmrText (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_POLYTEXTOUTA**. This value is 0x00000060.

**Bounds (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19), which specifies the bounding rectangle in logical units.

**iGraphicsMode (4 bytes):** An unsigned integer that specifies the current graphics mode, from the [GraphicsMode](#Section_fe60a5a523044b639949e11bfdbebae9) enumeration (section 2.1.16).

**exScale (4 bytes):** A FLOAT value that specifies the X scale from page units to .01mm units if graphics mode is **GM\_COMPATIBLE**.

**eyScale (4 bytes):** A FLOAT value that specifies the Y scale from page units to .01mm units if graphics mode is **GM\_COMPATIBLE**.

**cStrings (4 bytes):** An unsigned integer that specifies the number of **EmrText** objects.

**aEmrText (variable):** An array of [EmrText](#Section_dd585d0a5d7c4034963a1141af836972) objects (section 2.2.5) that specify the output strings in 8-bit ASCII characters, with text attributes, and spacing values. The number of EmrText objects is specified by **cStrings**.

The font and text colors used for output are specified by properties in the current state of EMF metafile playback (section [3.1](#Section_50ae16a09d1748018bf1e28e9533a889)).

EMR\_POLYTEXTOUTA SHOULD[<70>](#Appendix_A_70" \o "Product behavior note 70) be emulated with a series of [EMR\_EXTTEXTOUTW](#Section_a59a79ac328e492da34de02727af6edf) records (section [2.3.5.7](#Section_6b582a713c294fc6a0f41f8a313739a1)), one per string. This requires the ASCII text string in each EmrText object to be converted to [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) [**UTF16-LE**](#gt_f25550c9-f84f-4eb2-8156-14794a7e3059) encoding.

See section 2.3.5 for more drawing record types.

#### EMR\_POLYTEXTOUTW Record

The EMR\_POLYTEXTOUTW record draws one or more [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) text strings using the current font and text colors.

Fields not specified in this section are specified in section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| iGraphicsMode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| exScale | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| eyScale | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cStrings | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| wEmrText (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_POLYTEXTOUTW**. This value is 0x00000061.

**Bounds (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19), which specifies the bounding rectangle in logical units.

**iGraphicsMode (4 bytes):** An unsigned integer that specifies the current graphics mode. Graphics modes are specified in section [2.1.16](#Section_fe60a5a523044b639949e11bfdbebae9).

**exScale (4 bytes):** A FLOAT value that specifies the X scale from page units to .01mm units if graphics mode is **GM\_COMPATIBLE**.

**eyScale (4 bytes):** A FLOAT value that specifies the X scale from page units to .01mm units if graphics mode is **GM\_COMPATIBLE**.

**cStrings (4 bytes):** An unsigned integer that specifies the number of **EmrText** objects.

**wEmrText (variable):** An array of [EmrText objects (section 2.2.5)](#Section_dd585d0a5d7c4034963a1141af836972) that specify the output strings in Unicode [**UTF16-LE**](#gt_f25550c9-f84f-4eb2-8156-14794a7e3059) characters, with text attributes and spacing values. The number of EmrText objects is specified by **cStrings**.

The font and text colors used for output are specified by properties in the current state of the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45).

EMR\_POLYTEXTOUTW SHOULD be emulated with a series of [EMR\_EXTTEXTOUTW](#Section_a59a79ac328e492da34de02727af6edf) records (section [2.3.5.7](#Section_6b582a713c294fc6a0f41f8a313739a1)), one per string.[<71>](#Appendix_A_71" \o "Product behavior note 71)

See section 2.3.5 for more drawing record types.

#### EMR\_RECTANGLE Record

The EMR\_RECTANGLE record draws a rectangle. The rectangle is outlined by using the current pen and filled by using the current brush.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Box | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_RECTANGLE**. This value is 0x0000002B.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Box (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19), which specifies the [**inclusive-inclusive**](#gt_c706d1bc-8878-4743-9b94-e9643c5269e4) rectangle to draw.

The current drawing position is neither used nor updated by this record.

If a **PS\_NULL** pen is used, the dimensions of the rectangle are 1 pixel less in height and 1 pixel less in width.

See section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4) for more drawing record types.

#### EMR\_ROUNDRECT Record

The EMR\_ROUNDRECT record specifies a rectangle with rounded corners. The rectangle is outlined by using the current pen and filled by using the current brush.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Box | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Corner | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_ROUNDRECT**. This value is 0x0000002C.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Box (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19), which specifies the [**inclusive-inclusive**](#gt_c706d1bc-8878-4743-9b94-e9643c5269e4) bounding rectangle, in logical coordinates.

**Corner (8 bytes):** A 64-bit **SizeL** object ([MS-WMF] section 2.2.2.22), which specifies the **width** and **height**, in logical coordinates, of the ellipse used to draw the rounded corners.

See section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4) for more drawing record types.

#### EMR\_SETPIXELV Record

The EMR\_SETPIXELV record defines the color of the pixel at the specified logical coordinates.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Pixel | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Color | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_SETPIXELV**. This value is 0x0000000F.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Pixel (8 bytes):** A PointL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.15) that specifies the logical coordinates for the pixel.

**Color (4 bytes):** A 32-bit ColorRef object ([MS-WMF] section 2.2.2.8) that specifies the pixel color.

See section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4) for more drawing record types.

#### EMR\_SMALLTEXTOUT Record

The EMR\_SMALLTEXTOUT record outputs a string.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| x | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| y | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cChars | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| fuOptions | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| iGraphicsMode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| exScale | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| eyScale | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds (optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| TextString (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_SMALLTEXTOUT**. This value is 0x0000006C.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**x (4 bytes):** A signed integer specifying the x-coordinate of where to place the string.

**y (4 bytes):** A signed integer specifying the y-coordinate of where to place the string.

**cChars (4 bytes):** An unsigned integer specifying the number of 16-bit characters in the string. The string is NOT null-terminated.

**fuOptions (4 bytes):** An unsigned integer specifying the text output options to use. These options are specified by one or a combination of values from the [ExtTextOutOptions enumeration (section 2.1.11)](#Section_e7ffcc5340d148738edac5c5ee104aa5).

**iGraphicsMode (4 bytes):** An unsigned integer specifying the graphics mode, from the [GraphicsMode enumeration (section 2.1.16)](#Section_fe60a5a523044b639949e11bfdbebae9).

**exScale (4 bytes):** A FLOAT value that specifies how much to scale the text in the x-direction.

**eyScale (4 bytes):** A FLOAT value that specifies how much to scale the text in the y-direction.

**Bounds (16 bytes, optional):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19) that specifies the bounding rectangle in logical units.

**TextString (variable):** A string that contains the text string to draw, in either 8-bit or 16-bit character codes, according to the value of the **fuOptions** field.

If **ETO\_SMALL\_CHARS** is set in the **fuOptions** field, **TextString** contains 8-bit codes for characters, derived from the low bytes of [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) [**UTF16-LE**](#gt_f25550c9-f84f-4eb2-8156-14794a7e3059) character codes, in which the high byte is assumed to be 0.

If **ETO\_NO\_RECT** is set in the **fuOptions** field, the **Bounds** field is not included in the record.

See section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4) for more drawing record types.

#### EMR\_STROKEANDFILLPATH Record

The EMR\_STROKEANDFILLPATH record closes any open figures in a path, strokes the outline of the path by using the current pen, and fills its interior by using the current brush.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_STROKEANDFILLPATH**. This value is 0x0000003F.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Bounds (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19) that specifies the bounding rectangle in logical units.

See section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4) for more drawing record types.

#### EMR\_STROKEPATH Record

The EMR\_STROKEPATH record renders the specified path by using the current pen.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_STROKEPATH**. This value is 0x00000040.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Bounds (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19) that specifies the bounding rectangle in logical units.

See section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4) for more drawing record types.

### Escape Record Types

The **Escape** record types execute [**printer driver**](#gt_1a48eebd-e72c-494d-b8cb-84dfb7bc3b65) functions.

The following are [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) escape record types.

| Name | Section | Description |
| --- | --- | --- |
| EMR\_DRAWESCAPE | [2.3.6.1](#Section_86405bb1cc2c42bc9e0f28874f208d98) | Passes arbitrary information to the printer driver. The intent is that the information results in drawing being done. |
| EMR\_EXTESCAPE | [2.3.6.2](#Section_a31ef1e754ba41aeb79de6fd102dc6b1) | Passes arbitrary information to the printer driver. The intent is that the information does not result in drawing being done. |
| EMR\_NAMEDESCAPE | [2.3.6.3](#Section_125759e0c8394dc1b5fbe014e7e1159e) | Passes arbitrary information to the given named printer driver. |

The generic structure of escape records is specified as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| iEscape | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EscapeRecordBuffer (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that defines the type of the record. The escape record types are listed in the following table. See the preceding table for descriptions of these records.

| Name | Value |
| --- | --- |
| EMR\_DRAWESCAPE | 0x00000069 |
| EMR\_EXTESCAPE | 0x0000006A |
| EMR\_NAMEDESCAPE | 0x0000006E |

**Size (4 bytes):** An unsigned integer that specifies the size in bytes of this record in the [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7). This value MUST be a multiple of 4 bytes.

**iEscape (4 bytes):** An unsigned integer that specifies the printer driver escape to execute. This MUST be one of the values in the MetafileEscapes enumeration ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.1.1.17).

**EscapeRecordBuffer (variable):** An array of bytes that contains the remainder of the escape record.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| EscapeRecordParm (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| AlignmentPadding (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**EscapeRecordParm (variable):** An array of bytes that contains the parameters for the escape record.

**AlignmentPadding (variable, optional):** An array of up to 3 bytes that pads the record so that its total size is a multiple of 4 bytes. This field MUST be ignored.

See section [2.3](#Section_e0137630f3ad492cbde9e68866e255ba) for more EMF record types.

#### EMR\_DRAWESCAPE Record

The EMR\_DRAWESCAPE record passes arbitrary information to a [**printer driver**](#gt_1a48eebd-e72c-494d-b8cb-84dfb7bc3b65). The intent is that the information results in drawing being done.

Fields not specified in this section are specified in section [2.3.6](#Section_5370873a5fe64fdd8c5abade4ee54a1f).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| iEscape | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cjIn | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type from the RecordType enumeration (section [2.1.1](#Section_1eec80ba799b4784a9ac91597d590ae1)). It MUST be **EMR\_DRAWESCAPE**, which is 0x00000069.

**cjIn (4 bytes):** An unsigned integer specifying the number of bytes to pass to the printer driver.

**Data (variable):** The data to pass to the printer driver. There MUST be **cjIn** bytes available.

See section 2.3.6 for more escape record types.

#### EMR\_EXTESCAPE Record

The EMR\_EXTESCAPE record passes arbitrary information to a [**printer driver**](#gt_1a48eebd-e72c-494d-b8cb-84dfb7bc3b65). The intent is that the information does not result in drawing being done.

Fields not specified in this section are specified in section [2.3.6](#Section_5370873a5fe64fdd8c5abade4ee54a1f).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| iEscape | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cjIn | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type from the RecordType enumeration (section [2.1.1](#Section_1eec80ba799b4784a9ac91597d590ae1)). This value is 0x0000006A.

**cjIn (4 bytes):** An unsigned integer specifying the number of bytes to pass to the printer driver.

**Data (variable):** The data to pass to the printer driver. There MUST be **cjIn** bytes available.

See section 2.3.6 for more escape record types.

#### EMR\_NAMEDESCAPE Record

The **EMR\_NAMEDESCAPE** record passes arbitrary information to a named [**printer driver**](#gt_1a48eebd-e72c-494d-b8cb-84dfb7bc3b65).

Fields not specified in this section are specified in section [2.3.6](#Section_5370873a5fe64fdd8c5abade4ee54a1f).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| iEscape | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cjDriver | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cjIn | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DriverName (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type from the RecordType enumeration (section [2.1.1](#Section_1eec80ba799b4784a9ac91597d590ae1)). It MUST be EMR\_NAMEDESCAPE, which is 0x0000006E.

**cjDriver (4 bytes):** An unsigned integer that specifies the number of bytes in the **DriverName** field. This value MUST be an even number.

**cjIn (4 bytes):** An unsigned integer specifying the number of bytes in the **Data** field to pass to the printer driver.

**DriverName (variable):** A null-terminated string of [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) characters that specifies the name of the printer driver to receive data.

**Data (variable):** The data to pass to the printer driver.

See section 2.3.6 for more escape record types.

### Object Creation Record Types

The **Object Creation** record types create graphics objects.

The following are [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) object creation record types.

| Name | Section | Description |
| --- | --- | --- |
| EMR\_CREATEBRUSHINDIRECT | [2.3.7.1](#Section_b9a8ef5d00894e42b317e6ebc0ff098f) | Defines a logical brush with a LogBrushEx object (section [2.2.12](#Section_6d790f8d51654746bac33443915c2071)). |
| EMR\_CREATECOLORSPACE | [2.3.7.2](#Section_954af0ffa8d74d3480ed89f570bac016) | Defines a logical [**color space**](#gt_5d3fb5ea-c686-4d3b-b0ff-aef33fe1aee5) with a LogColorSpace object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.11). |
| EMR\_CREATECOLORSPACEW | [2.3.7.3](#Section_d1cd21477fb64d978f7ea289006b6064) | Defines a logical color space with a LogColorSpaceW object ([MS-WMF] section 2.2.2.12). |
| EMR\_CREATEDIBPATTERNBRUSHPT | [2.3.7.4](#Section_332116b4c6f94b18a7cc22c531b52afc) | Defines a pattern brush with a DeviceIndependentBitmap object ([MS-WMF] section 2.2.2.9). |
| EMR\_CREATEMONOBRUSH | [2.3.7.5](#Section_49b4227731b04eb9a6af86d9be9b568f) | Defines a monochrome pattern brush with a monochrome DeviceIndependentBitmap object. |
| EMR\_CREATEPALETTE | [2.3.7.6](#Section_07e1492be4bb4394934f4eaee67ab8ff) | Defines a [**logical palette**](#gt_43e451cc-1df4-4abc-bb24-aac32f5ab6f3) with a LogPalette object (section [2.2.17](#Section_758f047d765e424c9204a833b7b4e527)). |
| EMR\_CREATEPEN | [2.3.7.7](#Section_2374647fdf6748e386aa384715c28e71) | Defines a logical pen with a LogPen object (section [2.2.19](#Section_93ce3f4537ac4affb6e82f6db054c4c4)). |
| EMR\_EXTCREATEFONTINDIRECTW | [2.3.7.8](#Section_7e266b6d32e54201b6878ec40c24cd73) | Defines a logical font with either a LogFont object (section [2.2.13](#Section_1cd7cb7a1a7c48d2b4a4c218f9d12100)) or LogFontExDv object (section [2.2.15](#Section_595fcaaebc3741489afe859e81ca7f76)). |
| EMR\_EXTCREATEPEN | [2.3.7.9](#Section_d7f51e054024497cad4a8aeca9d34256) | Defines a logical pen with a LogPenEx object (section [2.2.20](#Section_5b67b3eeea004f809b732959804381be)) and optional DeviceIndependentBitmap object. |

The generic structure of object creation records is specified as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ObjectRecordBuffer (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that defines the type of record. The object creation record types are listed in the following table. See the preceding table for descriptions of these record types.

| Name | Value |
| --- | --- |
| EMR\_CREATEMONOBRUSH | 0x0000005D |
| EMR\_CREATEDIBPATTERNBRUSHPT | 0x0000005E |
| EMR\_EXTCREATEPEN | 0x0000005F |
| EMR\_CREATECOLORSPACEW | 0x0000007A |
| EMR\_CREATEPEN | 0x00000026 |
| EMR\_CREATEBRUSHINDIRECT | 0x00000027 |
| EMR\_CREATEPALETTE | 0x00000031 |
| EMR\_EXTCREATEFONTINDIRECTW | 0x00000052 |
| EMR\_CREATECOLORSPACE | 0x00000063 |

**Size (4 bytes):** An unsigned integer that specifies the size in bytes of this record in the [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7). This value MUST be a multiple of 4 bytes.

**ObjectRecordBuffer (variable):** An array of bytes that contains the remainder of the object creation record.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| ObjectRecordParm (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| AlignmentPadding (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**ObjectRecordParm (variable):** An array of bytes that contains the parameters for the object creation record.

**AlignmentPadding (variable, optional):** An array of up to 3 bytes that pads the record so that its total size is a multiple of 4 bytes. This field MUST be ignored.

See section [2.3](#Section_e0137630f3ad492cbde9e68866e255ba) for more EMF record types.

#### EMR\_CREATEBRUSHINDIRECT Record

The **EMR\_CREATEBRUSHINDIRECT** record defines a logical brush for graphics operations.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ihBrush | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| LogBrush | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as EMR\_CREATEBRUSHINDIRECT. This value is 0x00000027.

**Size (4 bytes):** An unsigned integer that specifies the size in bytes, of this record. This value is 0x00000018.

**ihBrush (4 bytes):** An unsigned integer that specifies the index of the logical brush object in the EMF object table (section [3.1.1.1](#Section_e4fa4e6390964cdcb77685e2a1e4e1f4)). This index is used to refer to the object, so it can be reused or modified.

**LogBrush (12 bytes):** A LogBrushEx object (section [2.2.12](#Section_6d790f8d51654746bac33443915c2071)) that specifies the style, color, and pattern of the logical brush. The **BrushStyle** field in this object MUST be BS\_SOLID, BS\_HATCHED, or BS\_NULL.

See section [2.3.7](#Section_7d914464d5464393bddb6d1e747fca18) for more object creation record types.

#### EMR\_CREATECOLORSPACE Record

The **EMR\_CREATECOLORSPACE** record creates a logical [**color space**](#gt_5d3fb5ea-c686-4d3b-b0ff-aef33fe1aee5) object from a [**color profile**](#gt_cffe5dfa-5408-4302-8ab0-400cb94f5e32) with a name consisting of [**ASCII**](#gt_79fa85ca-ac61-467c-b819-e97dc1a7a599) characters.[<72>](#Appendix_A_72" \o "Product behavior note 72)

Fields not specified in this section are specified in section [2.3.7](#Section_7d914464d5464393bddb6d1e747fca18).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ihCS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| lcs (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_CREATECOLORSPACE**. This value is 0x00000063.

**ihCS (4 bytes):** An unsigned integer that specifies the index of the logical color space object in the **EMF object table** (section [3.1.1.1](#Section_e4fa4e6390964cdcb77685e2a1e4e1f4)). This index MUST be saved so that this object can be reused or modified.

**lcs (variable):** A LogColorSpace object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.11), which can specify the name of a color profile in ASCII characters.

The logical color space object defined by this record can be selected into the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45) by an EMR\_SETCOLORSPACE record (section [2.3.8.7](#Section_2a84d7a5f8c14dd2ae79a029a25ad601)), which defines the logical color space to use in subsequent graphics operations.

See section 2.3.7 for more object creation record types.

#### EMR\_CREATECOLORSPACEW Record

The EMR\_CREATECOLORSPACEW record creates a logical [**color space**](#gt_5d3fb5ea-c686-4d3b-b0ff-aef33fe1aee5) object from a [**color profile**](#gt_cffe5dfa-5408-4302-8ab0-400cb94f5e32) with a name consisting of [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) characters.[<73>](#Appendix_A_73" \o "Product behavior note 73)

Fields not specified in this section are specified in section [2.3.7](#Section_7d914464d5464393bddb6d1e747fca18).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ihCS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| lcs (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwFlags | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbData | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Data (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_CREATECOLORSPACEW**. This value is 0x0000007A.

**ihCS (4 bytes):** An unsigned integer that specifies the index of the logical color space object in the **EMF object table** (section [3.1.1.1](#Section_e4fa4e6390964cdcb77685e2a1e4e1f4)). This index MUST be saved so that this object can be reused or modified.

**lcs (variable):** A LogColorSpaceW object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.12) that can specify the name of a color profile in Unicode [**UTF16-LE**](#gt_f25550c9-f84f-4eb2-8156-14794a7e3059) characters.

**dwFlags (4 bytes):** An unsigned integer that provides information about the data in this record.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | C |

**C (1 bit):** If set, the **Data** field contains color profile data.

**cbData (4 bytes):** An unsigned integer that specifies the size in bytes, of the **Data** field.

**Data (variable, optional):** An array of bytes that specifies color profile data. When **cbData** is zero, this field is optional and is ignored.

The logical color space object defined by this record can be selected into the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45) by an EMR\_SETCOLORSPACE record (section [2.3.8.7](#Section_2a84d7a5f8c14dd2ae79a029a25ad601)), which defines the logical color spaceto use in subsequent graphics operations.

See section 2.3.7 for more object creation record types.

#### EMR\_CREATEDIBPATTERNBRUSHPT Record

The EMR\_CREATEDIBPATTERNBRUSHPT record defines a pattern brush for graphics operations. The pattern is specified by a [**DIB**](#gt_3a6934d5-cc59-4ab9-a54e-e062a09a4091).

Fields not specified in this section are specified in section [2.3.7](#Section_7d914464d5464393bddb6d1e747fca18).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ihBrush | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Usage | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offBmi | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbBmi | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offBits | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbBits | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BitmapBuffer (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as EMR\_CREATEDIBPATTERNBRUSHPT. This value is 0x0000005E.

**ihBrush (4 bytes):** An unsigned integer that specifies the index of the pattern brush object in the **EMF object table** (section [3.1.1.1](#Section_e4fa4e6390964cdcb77685e2a1e4e1f4)). This index MUST be saved so that this object can be reused or modified.

**Usage (4 bytes):** An unsigned integer that specifies how to interpret values in the [**color table**](#gt_4737483c-820c-41e2-a7be-0ffed2eb4418) in the DIB header. This value is in the DIBColors enumeration (section [2.1.9](#Section_a5e722e3891a4a67be1aed5a48a7fda1)).

**offBmi (4 bytes):** An unsigned integer that specifies the offset from the start of this record to the DIB header.

**cbBmi (4 bytes):** An unsigned integer that specifies the size of the DIB header.

**offBits (4 bytes):** An unsigned integer that specifies the offset from the start of this record to the DIB bits.

**cbBits (4 bytes):** An unsigned integer that specifies the size of the DIB bits.

**BitmapBuffer (variable):** A buffer containing a [**packed DIB**](#gt_6b4f9d56-b04f-4fd3-a069-412866dcce4d) in the form of a DeviceIndependentBitmap object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.9). It is not required to be contiguous with the fixed portion of this record.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| UndefinedSpace (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BmiSrc (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BitsSrc (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**UndefinedSpace (variable, optional):** An array of bytes that MUST be ignored.

**BmiSrc (variable):** The DIB header, which is the **DibHeaderInfo** field of a DeviceIndependentBitmap object.

**BitsSrc (variable):** The DIB bits, which is the **aData** field of a DeviceIndependentBitmap object.

The pattern brush object defined by this record can be selected into the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45) by an EMR\_SELECTOBJECT record (section [2.3.8.5](#Section_145b063d5f9641feb7ae1e615b2bc2bf)), which specifies the pattern brush to use in subsequent graphics operations.

See section 2.3.7 for more object creation record types.

#### EMR\_CREATEMONOBRUSH Record

The **EMR\_CREATEMONOBRUSH** record defines a monochrome pattern brush for graphics operations. The pattern is specified by a monochrome [**DIB**](#gt_3a6934d5-cc59-4ab9-a54e-e062a09a4091).

Fields not specified in this section are specified in section [2.3.7](#Section_7d914464d5464393bddb6d1e747fca18).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ihBrush | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Usage | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offBmi | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbBmi | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offBits | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbBits | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BitmapBuffer (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_CREATEMONOBRUSH**. This value is 0x0000005D.

**ihBrush (4 bytes):** An unsigned integer that specifies the index of the monochrome pattern brush object in the EMF object table (section [3.1.1.1](#Section_e4fa4e6390964cdcb77685e2a1e4e1f4)). This index MUST be saved so that this object can be reused or modified.

**Usage (4 bytes):** An unsigned integer that specifies how to interpret values in the [**color table**](#gt_4737483c-820c-41e2-a7be-0ffed2eb4418) in the DIB header. This value is in the [DIBColors](#Section_a5e722e3891a4a67be1aed5a48a7fda1) enumeration (section 2.1.9).

**offBmi (4 bytes):** An unsigned integer that specifies the offset from the start of this record to the DIB header.

**cbBmi (4 bytes):** An unsigned integer that specifies the size of the DIB header.

**offBits (4 bytes):** An unsigned integer that specifies the offset from the start of this record to the DIB bits.

**cbBits (4 bytes):** An unsigned integer that specifies the size of the DIB bits.

**BitmapBuffer (variable):** A buffer containing a [**packed DIB**](#gt_6b4f9d56-b04f-4fd3-a069-412866dcce4d) in the form of a monochrome DeviceIndependentBitmap object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.9). It is not required to be contiguous with the fixed portion of this record.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| UndefinedSpace (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BmiSrc (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BitsSrc (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**UndefinedSpace (variable, optional):** An array of bytes that MUST be ignored.

**BmiSrc (variable):** The DIB header, which is the **DibHeaderInfo** field of a DeviceIndependentBitmap object.

**BitsSrc (variable):** The DIB bits, which is the **aData** field of a DeviceIndependentBitmap object.

The monochrome pattern brush object defined by this record can be selected into the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45) by an EMR\_SELECTOBJECT record (section [2.3.8.5](#Section_145b063d5f9641feb7ae1e615b2bc2bf)), which specifies the pattern brush to use in subsequent graphics operations.

See section 2.3.7 for more object creation record types.

#### EMR\_CREATEPALETTE Record

The EMR\_CREATEPALETTE record defines a [**logical palette**](#gt_43e451cc-1df4-4abc-bb24-aac32f5ab6f3) for graphics operations.

Fields not specified in this section are specified in section [2.3.7](#Section_7d914464d5464393bddb6d1e747fca18).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ihPal | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| LogPalette (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_CREATEPALETTE**. This value is 0x00000031.

**ihPal (4 bytes):** An unsigned integer that specifies the index of the logical palette object in the EMF object table (section [3.1.1.1](#Section_e4fa4e6390964cdcb77685e2a1e4e1f4)). This index MUST be saved so that this object can be reused or modified.

**LogPalette (variable):** A [LogPalette](#Section_758f047d765e424c9204a833b7b4e527) object (section 2.2.17). The **Version** field of this object MUST be set to 0x0300. If the **NumberOfEntries** value in this object is zero, processing of this record MUST fail.

The logical palette defined by this record can be selected into the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45) by an [EMR\_SELECTPALETTE](#Section_e6a4ce2a209d43dfb7635d8e54c21a10) record (section 2.3.8.6), which specifies the logical palette to use in subsequent graphics operations.

See section 2.3.7 for more object creation record types.

#### EMR\_CREATEPEN Record

The EMR\_CREATEPEN record defines a logical pen for graphics operations.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ihPen | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| LogPen | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_CREATEPEN**. This value is 0x00000026.

**Size (4 bytes):** An unsigned integer that specifies the size in bytes, of this record. This value is 0x0000001C.

**ihPen (4 bytes):** An unsigned integer that specifies the index of the logical pen object in the EMF object table (section [3.1.1.1](#Section_e4fa4e6390964cdcb77685e2a1e4e1f4)). This index MUST be saved so that this object can be reused or modified.

**LogPen (16 bytes):** A [LogPen](#Section_93ce3f4537ac4affb6e82f6db054c4c4) object (section 2.2.19) that specifies the style, width, and color of the logical pen.

The logical pen object defined by this record can be selected into the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45) by an [EMR\_SELECTOBJECT](#Section_145b063d5f9641feb7ae1e615b2bc2bf) record (section 2.3.8.5), which specifies the logical pen to use in subsequent graphics operations.

See section [2.3.7](#Section_7d914464d5464393bddb6d1e747fca18) for more object creation record types.

#### EMR\_EXTCREATEFONTINDIRECTW Record

The **EMR\_EXTCREATEFONTINDIRECTW** record defines a logical font for graphics operations.

Fields not specified in this section are specified in section [2.3.7](#Section_7d914464d5464393bddb6d1e747fca18).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ihFonts | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| elw (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_EXTCREATEFONTINDIRECTW**. This value is 0x00000052.

**ihFonts (4 bytes):** An unsigned integer that specifies the index of the logical font object in the **EMF object table** (section [3.1.1.1](#Section_e4fa4e6390964cdcb77685e2a1e4e1f4)). This index MUST be saved so that this object can be reused or modified.

**elw (variable):** A LogFontExDv object (section [2.2.15](#Section_595fcaaebc3741489afe859e81ca7f76)), which specifies the logical font. A LogFont object (section [2.2.13](#Section_1cd7cb7a1a7c48d2b4a4c218f9d12100)) MAY[<74>](#Appendix_A_74" \o "Product behavior note 74) be present instead. The process for determining the type of object in this field is described below.

The logical font object defined by this record can be selected into the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45) by an EMR\_SELECTOBJECT record (section [2.3.8.5](#Section_145b063d5f9641feb7ae1e615b2bc2bf)), which specifies the logical font to use in subsequent graphics operations.

The type of logical font object in the **elw** field of this record is determined by the following algorithm (all size and length values are in bytes):

* First, note that the size in bytes of the static part of this record—that is, the sum of the sizes of its **Type**, **Size**, and **ihFonts** fields—is 12.
* Next, note that because the size in bytes of the entire record is present in its **Size** field, the size in bytes of the variable-length **elw** field can be computed as follows.

1. Size - 12

* The size of the **elw** field must be equal to or greater than the size of a **LogFontPanose** object.[<75>](#Appendix_A_75" \o "Product behavior note 75)
* If the size of the **elw** field is equal to the size of a **LogFontPanose** object (section [2.2.16](#Section_c68ffff81c324398ac812cb6f0cd7a7c)), **elw** MUST be treated as a fixed-length **LogFontPanose** object.
* If the size of the **elw** field is greater than the size of a **LogFontPanose** object, then **elw** MUST be treated as a variable-length **LogFontExDv** object.

The size of a LogFontPanose object is 0x0140 (320 decimal). It is determined by adding up the sizes of its fields, as follows:

* **LogFont**: The size of a LogFont object is 0x005C (92 decimal). It is determined by adding up the sizes of its fields, as follows:
  + Fields from **Height** through **PitchAndFamily**: 0x001C (28 decimal).
  + **Facename**: The length is 32 16-bit characters: 0x0040 (64 decimal).
* **Fullname**: The length is 64 16-bit characters: 0x0080 (128 decimal).
* **Style**: The length is 32 16-bit characters: 0x0040 (64 decimal).
* Fields from **Version** through **Culture**: 0x0018 (24 decimal).
* **Panose**: The exact length of this field is 0x000A, but it MUST be padded by two additional bytes for 32-bit alignment, so for the purposes of this computation the length is 0x000C (12 decimal).

See section 2.3.7 for more object creation record types.

#### EMR\_EXTCREATEPEN Record

The **EMR\_EXTCREATEPEN** record defines an extended logical pen for graphics operations. An optional [**DIB**](#gt_3a6934d5-cc59-4ab9-a54e-e062a09a4091) can be specified to use as the line style.

Fields not specified in this section are specified in section [2.3.7](#Section_7d914464d5464393bddb6d1e747fca18).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ihPen | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offBmi | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbBmi | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offBits | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbBits | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| elp (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BitmapBuffer (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_EXTCREATEPEN**. This value is 0x0000005F.

**ihPen (4 bytes):** An unsigned integer that specifies the index of the extended logical pen object in the **EMF object table** (section [3.1.1.1](#Section_e4fa4e6390964cdcb77685e2a1e4e1f4)). This index MUST be saved so that this object can be reused or modified.

**offBmi (4 bytes):** An unsigned integer that specifies the offset from the start of this record to the DIB header if the record contains a DIB.

**cbBmi (4 bytes):** An unsigned integer that specifies the size of the DIB header if the record contains a DIB.

**offBits (4 bytes):** An unsigned integer that specifies the offset from the start of this record to the DIB bits if the record contains a DIB.

**cbBits (4 bytes):** An unsigned integer that specifies the size of the DIB bits if the record contains a DIB.

**elp (variable):** A LogPenEx object (section [2.2.20](#Section_5b67b3eeea004f809b732959804381be)) that specifies an extended logical pen with attributes including an optional line style array.

**BitmapBuffer (variable, optional):** An array of bytes containing a [**packed DIB**](#gt_6b4f9d56-b04f-4fd3-a069-412866dcce4d) in the form of a DeviceIndependentBitmap object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.9). It is not required to be contiguous with the fixed portion of this record.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| UndefinedSpace (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BmiSrc (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BitsSrc (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**UndefinedSpace (variable, optional):** An array of bytes that MUST be ignored.

**BmiSrc (variable):** The DIB header, which is the **DibHeaderInfo** field of a DeviceIndependentBitmap object.

**BitsSrc (variable):** The DIB bits, which is the **aData** field of a DeviceIndependentBitmap object.

The extended logical pen object defined by this record can be selected into the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45) by an EMR\_SELECTOBJECT record (section [2.3.8.5](#Section_145b063d5f9641feb7ae1e615b2bc2bf)), which specifies the logical pen to use in subsequent graphics operations.

See section 2.3.7 for more object creation record types.

### Object Manipulation Record Types

The **Object Manipulation** record types manage and modify graphics objects.

The following are [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) object manipulation record types.

| Name | Section | Description |
| --- | --- | --- |
| EMR\_COLORCORRECTPALETTE | [2.3.8.1](#Section_99de1bfb3d77455b86797386903e1472) | Specifies how to correct the entries of a LogPalette object (section [2.2.17](#Section_758f047d765e424c9204a833b7b4e527)) by using [**WCS**](#gt_e4f9c4a4-c195-43d0-b189-f9727ec8a2b3) values. |
| EMR\_DELETECOLORSPACE | [2.3.8.2](#Section_5d137387d79a4bc89a4d38291320e148) | Specifies how to delete a logical color space from the EMF object table (section [3.1.1.1](#Section_e4fa4e6390964cdcb77685e2a1e4e1f4)). |
| EMR\_DELETEOBJECT | [2.3.8.3](#Section_6f0f12a3111a478b8251a9505168f9a9) | Specifies the index of the object to be deleted from the EMF object table. |
| EMR\_RESIZEPALETTE | [2.3.8.4](#Section_ec78933296b54bb89d7c8b5c8c0da8b9) | Increases or decreases the size of an existing LogPalette object. |
| EMR\_SELECTOBJECT | [2.3.8.5](#Section_145b063d5f9641feb7ae1e615b2bc2bf) | Specifies an existing object based on its index in the EMF object table and selects it into the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45) |
| EMR\_SELECTPALETTE | [2.3.8.6](#Section_e6a4ce2a209d43dfb7635d8e54c21a10) | Selects the specified LogPalette object into the playback device context. |
| EMR\_SETCOLORSPACE | [2.3.8.7](#Section_2a84d7a5f8c14dd2ae79a029a25ad601) | Specifies a logical color space, based on its index in the EMF object table. |
| EMR\_SETPALETTEENTRIES | [2.3.8.8](#Section_883482963c9a488fbbf719c897535372) | Defines [**RGB**](#gt_2c716d3a-e60b-4e52-bbb0-2fdeb298003b) color values in a range of entries for an existing LogPalette object. |

The generic structure of object manipulation records is specified as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ObjectRecordBuffer (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that defines the type of record. The object manipulation record types are listed in the following table. See the preceding table for descriptions of these record types.

| Name | Value |
| --- | --- |
| EMR\_SELECTOBJECT | 0x00000025 |
| EMR\_DELETEOBJECT | 0x00000028 |
| EMR\_SELECTPALETTE | 0x00000030 |
| EMR\_SETPALETTEENTRIES | 0x00000032 |
| EMR\_RESIZEPALETTE | 0x00000033 |
| EMR\_SETCOLORSPACE | 0x00000064 |
| EMR\_DELETECOLORSPACE | 0x00000065 |
| EMR\_COLORCORRECTPALETTE | 0x0000006F |

**Size (4 bytes):** An unsigned integer that specifies the size in bytes of this record in the [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7). This value MUST be a multiple of 4 bytes.

**ObjectRecordBuffer (variable):** An array of bytes that contains the remainder of the object manipulation record.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| ObjectRecordParm (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| AlignmentPadding (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**ObjectRecordParm (variable):** An array of bytes that contains the parameters for the object manipulation record.

**AlignmentPadding (variable, optional):** An array of up to 3 bytes that pads the record so that its total size is a multiple of 4 bytes. This field MUST be ignored.

See section [2.3](#Section_e0137630f3ad492cbde9e68866e255ba) for more EMF record types.

#### EMR\_COLORCORRECTPALETTE Record

The **EMR\_COLORCORRECTPALETTE** record specifies the correction of entries of a [**logical palette**](#gt_43e451cc-1df4-4abc-bb24-aac32f5ab6f3) object using [**WCS**](#gt_e4f9c4a4-c195-43d0-b189-f9727ec8a2b3).[<76>](#Appendix_A_76" \o "Product behavior note 76)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ihPalette | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| nFirstEntry | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| nPalEntries | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| nReserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_COLORCORRECTPALETTE**. This value is 0x0000006F.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes. This value is 0x00000018.

**ihPalette (4 bytes):** An unsigned integer that specifies the index of a logical palette object (section [2.2.17](#Section_758f047d765e424c9204a833b7b4e527)) in the **EMF object table** (section [3.1.1.1](#Section_e4fa4e6390964cdcb77685e2a1e4e1f4)).

**nFirstEntry (4 bytes):** An unsigned integer that specifies the index of the first entry to correct.

**nPalEntries (4 bytes):** An unsigned integer that specifies the number of [**palette**](#gt_dca5cd91-8d01-424b-9722-c9e818562ee5) entries to correct.

**nReserved (4 bytes):** An unsigned integer that is undefined and unused.

See section [2.3.8](#Section_2f1c3268b43b4c28988b41aa49354998) for more object manipulation record types.

#### EMR\_DELETECOLORSPACE Record

The **EMR\_DELETECOLORSPACE** record deletes a logical [**color space**](#gt_5d3fb5ea-c686-4d3b-b0ff-aef33fe1aee5) object.[<77>](#Appendix_A_77" \o "Product behavior note 77)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ihCS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_DELETECOLORSPACE**. This value is 0x00000065.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**ihCS (4 bytes):** An unsigned integer that specifies the index of a logical color space object in the EMF object table (section [3.1.1.1](#Section_e4fa4e6390964cdcb77685e2a1e4e1f4)).

The color space is specified by either a LogColorSpace or LogColorSpaceW object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) sections 2.2.2.11 and 2.2.2.12, respectively). If the deleted color space is currently selected into the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45), the default object MUST be restored.

An EMR\_DELETEOBJECT record (section [2.3.8.3](#Section_6f0f12a3111a478b8251a9505168f9a9)) SHOULD be used instead of this record to delete a logical color space object.

See section [2.3.8](#Section_2f1c3268b43b4c28988b41aa49354998) for more object manipulation record types.

#### EMR\_DELETEOBJECT Record

The **EMR\_DELETEOBJECT** record deletes a graphics object, which is specified by its index in the EMF object table (section [3.1.1.1](#Section_e4fa4e6390964cdcb77685e2a1e4e1f4)).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ihObject | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_DELETEOBJECT**. This value is 0x00000028.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**ihObject (4 bytes):** An unsigned integer that specifies the index of a graphics object in the EMF object table.

This value MUST NOT be 0, which is a reserved index that refers to the [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7) itself; and it MUST NOT be the index of a [**stock object**](#gt_b1328cb9-acd8-47f8-9896-fe985af2d6d6), which cannot be deleted. Stock object indexes are specified in the StockObject (section [2.1.31](#Section_d6dffd25861542f8aed1309f1fe54ab2)) enumeration.

The object specified by this record MUST be deleted from the EMF object table. If the deleted object is currently selected into the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45), the default object MUST be restored.

See section [2.3.8](#Section_2f1c3268b43b4c28988b41aa49354998) for more object manipulation record types.

#### EMR\_RESIZEPALETTE Record

The **EMR\_RESIZEPALETTE** record increases or decreases the size of an existing LogPalette object (section [2.2.17](#Section_758f047d765e424c9204a833b7b4e527)).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ihPal | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NumberOfEntries | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_RESIZEPALETTE**. This value is 0x00000033.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**ihPal (4 bytes):** An unsigned integer that specifies the index of the [**palette**](#gt_dca5cd91-8d01-424b-9722-c9e818562ee5) object in the EMF object table (section [3.1.1.1](#Section_e4fa4e6390964cdcb77685e2a1e4e1f4)).

**NumberOfEntries (4 bytes):** An unsigned integer that specifies the number of entries in the palette after resizing. The value MUST be <= 0x00000400 and > 0x00000000.[<78>](#Appendix_A_78" \o "Product behavior note 78)

The new size of the LogPalette object MUST be reflected in the **NumberOfEntries** field in that structure.

See section [2.3.8](#Section_2f1c3268b43b4c28988b41aa49354998) for more object manipulation record types.

#### EMR\_SELECTOBJECT Record

The **EMR\_SELECTOBJECT** record selects a graphics object into the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ihObject | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_SELECTOBJECT**. This value is 0x00000025.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**ihObject (4 bytes):** An unsigned integer that specifies either the index of a graphics object in the EMF object table (section [3.1.1.1](#Section_e4fa4e6390964cdcb77685e2a1e4e1f4)) or the index of a [**stock object**](#gt_b1328cb9-acd8-47f8-9896-fe985af2d6d6) in the StockObject enumeration (section [2.1.31](#Section_d6dffd25861542f8aed1309f1fe54ab2)).

The object index MUST NOT be zero, which is reserved and refers to the [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7) itself.

The object specified by this record MUST be used in subsequent EMF drawing operations, until another EMR\_SELECTOBJECT record changes the object of that type or the object is deleted.

See section [2.3.8](#Section_2f1c3268b43b4c28988b41aa49354998) for more object manipulation record types.

#### EMR\_SELECTPALETTE Record

The **EMR\_SELECTPALETTE** record selects a [**logical palette**](#gt_43e451cc-1df4-4abc-bb24-aac32f5ab6f3) into the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ihPal | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as EMR\_SELECTPALETTE. This value is 0x00000030.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes. This value is 0x0000000C.

**ihPal (4 bytes):** An unsigned integer that specifies either the index of a LogPalette object (section [2.2.17](#Section_758f047d765e424c9204a833b7b4e527)) in the [EMF object table](#Section_e4fa4e6390964cdcb77685e2a1e4e1f4) (section 3.1.1.1) or the value DEFAULT\_PALETTE from the StockObject enumeration (section [2.1.31](#Section_d6dffd25861542f8aed1309f1fe54ab2)), which is the index of a stock [**palette**](#gt_dca5cd91-8d01-424b-9722-c9e818562ee5).

The object index MUST NOT be zero, which is reserved and refers to the [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7) itself.

The palette specified by this record MUST be used in subsequent EMF drawing operations, until another EMR\_SELECTPALETTE record changes the object or the object is deleted.

See section [2.3.8](#Section_2f1c3268b43b4c28988b41aa49354998) for more object manipulation record types.

#### EMR\_SETCOLORSPACE Record

The **EMR\_SETCOLORSPACE** record selects a logical [**color space**](#gt_5d3fb5ea-c686-4d3b-b0ff-aef33fe1aee5) into the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45).[<79>](#Appendix_A_79" \o "Product behavior note 79)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ihCS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_SETCOLORSPACE**. This value is 0x00000064.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes. This value is 0x0000000C.

**ihCS (4 bytes):** An unsigned integer that specifies the index of a logical color space object in the EMF object table (section [3.1.1.1](#Section_e4fa4e6390964cdcb77685e2a1e4e1f4)).

This object is either a LogColorSpace or LogColorSpaceW object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) sections 2.2.2.11 and 2.2.2.12, respectively).

The color space specified by this record MUST be used in subsequent EMF drawing operations, until another EMR\_SETCOLORSPACE record changes the object or the object is deleted.

See section [2.3.8](#Section_2f1c3268b43b4c28988b41aa49354998) for more object manipulation record types.

#### EMR\_SETPALETTEENTRIES Record

The **EMR\_SETPALETTEENTRIES** record defines [**RGB**](#gt_2c716d3a-e60b-4e52-bbb0-2fdeb298003b) color values in a range of entries for an existing [**logical palette**](#gt_43e451cc-1df4-4abc-bb24-aac32f5ab6f3).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ihPal | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Start | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NumberofEntries | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| aPalEntries (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as EMR\_SETPALETTEENTRIES. This value is 0x00000032.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**ihPal (4 bytes):** An unsigned integer that specifies an index of a LogPalette object (section [2.2.17](#Section_758f047d765e424c9204a833b7b4e527)) in the EMF object table (section [3.1.1.1](#Section_e4fa4e6390964cdcb77685e2a1e4e1f4)).

**Start (4 bytes):** An unsigned integer that specifies the index in the palette of the first entry to set.

**NumberofEntries (4 bytes):** An unsigned integer that specifies the number of entries in the **aPalEntries** array.

**aPalEntries (variable):** An array of LogPaletteEntry objects (section [2.2.18](#Section_c1f7b285be164112a4e60b2fd4c1d148)) that specify the [**palette**](#gt_dca5cd91-8d01-424b-9722-c9e818562ee5) data.

See section [2.3.8](#Section_2f1c3268b43b4c28988b41aa49354998) for more object manipulation record types.

### OpenGL Record Types

The **OpenGL** record types specify [**OpenGL**](#gt_216becb9-a800-4a44-8206-2ec2231c9c67) functions [[OPENGL]](https://go.microsoft.com/fwlink/?LinkId=90241).

The following are [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) OpenGL record types.[<80>](#Appendix_A_80" \o "Product behavior note 80)

| Name | Section | Description |
| --- | --- | --- |
| EMR\_GLSBOUNDEDRECORD | [2.3.9.1](#Section_cb5036dac79d4aaeb1c0a48361447a0b) | Specifies an OpenGL function with a bounding rectangle for output. |
| EMR\_GLSRECORD | [2.3.9.2](#Section_4f2e25120c5d468bbb81484cb1d4aa89) | Specifies an OpenGL function. |

The generic structure of OpenGL records is specified as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OpenGLRecordBuffer (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that defines the type of record. The OpenGL record types are listed in the following table. See the preceding table for descriptions of these record types.

| Name | Value |
| --- | --- |
| EMR\_GLSRECORD | 0x00000066 |
| EMR\_GLSBOUNDEDRECORD | 0x00000067 |

**Size (4 bytes):** An unsigned integer that specifies the size in bytes of this record in the [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7). This value MUST be a multiple of 4 bytes.

**OpenGLRecordBuffer (variable):** An array of bytes that contains the remainder of the OpenGL record.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| OpenGLRecordParm (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| AlignmentPadding (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**OpenGLRecordParm (variable):** An array of bytes that contains the parameters for the OpenGL record.

**AlignmentPadding (variable, optional):** An array of up to 3 bytes that pads the record so that its total size is a multiple of 4 bytes. This field MUST be ignored.

See section [2.3](#Section_e0137630f3ad492cbde9e68866e255ba) for more EMF record types.

#### EMR\_GLSBOUNDEDRECORD Record

The **EMR\_GLSBOUNDEDRECORD** record specifies an [**OpenGL**](#gt_216becb9-a800-4a44-8206-2ec2231c9c67) function with a bounding rectangle for output.

Fields not specified in this section are specified in section [2.3.1](#Section_2a57d4bdf3dc45df95fd447382b5262a).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbData | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Data (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as EMR\_GLSBOUNDEDRECORD. This value is 0x00000067.

**Bounds (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19) that defines a bounding rectangle in logical units, for output produced by executing the OpenGL function.

**cbData (4 bytes):** An unsigned integer that specifies the size of the **Data** field in bytes. If this value is zero, no data is attached to this record.

**Data (variable, optional):** An array of bytes that specifies data for the OpenGL function.

See section [2.3.9](#Section_4a6c2fc239af43b48254449f5f1c5c05) for more OpenGL record types.

#### EMR\_GLSRECORD Record

The **EMR\_GLSRECORD** record specifies an [**OpenGL**](#gt_216becb9-a800-4a44-8206-2ec2231c9c67) function.

Fields not specified in this section are specified in section [2.3.1](#Section_2a57d4bdf3dc45df95fd447382b5262a).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbData | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Data (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as EMR\_GLSRECORD. This value is 0x00000066.

**cbData (4 bytes):** An unsigned integer that specifies the size in bytes, of the **Data** field. If this value is zero, no data is attached to this record.

**Data (variable, optional):** An array of bytes that specifies data for the OpenGL function.

See section [2.3.9](#Section_4a6c2fc239af43b48254449f5f1c5c05) for more OpenGL record types.

### Path Bracket Record Types

The **Path Bracket** record types are used to construct a [**path bracket**](#gt_b89ea7da-c0bf-4c45-9f3e-fc430d7ab601), which defines the current [**path**](#gt_2cd71385-2d9c-4ab8-bf4a-7b258816d613) in the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45).

**Note:** None of the path bracket records specify parameters.

The following are the path bracket record types.

| Name | Description |
| --- | --- |
| EMR\_ABORTPATH | This record closes path bracket construction and discards the current path. |
| EMR\_BEGINPATH | This record opens path bracket construction.  Once path bracket construction is open, an application can begin specifying records to define the points that lie in the path. Path bracket construction MUST be closed by an EMR\_ABORTPATH or EMR\_ENDPATH record.  When an application processes an EMR\_BEGINPATH record, path bracket construction MUST NOT be open. |
| EMR\_CLOSEFIGURE | This record closes the figure in path bracket construction.  Processing the EMR\_CLOSEFIGURE record closes the figure by drawing a line from the current drawing position to the first point of the figure, and then it connects the lines by using the current [**line join**](#gt_4fd3ebb8-54f8-47e8-aded-399a08ad873e). If the figure is closed by processing an EMR\_LINETO record (section [2.3.5.13](#Section_9b3eccf94a554a9cb0df3c495e7b9a8c)) instead of this record, the current [**line cap**](#gt_b0341150-ca76-4424-88fe-607e03848dbf) is used to create the corner instead of the line join. The line parameters are specified by the **PenStyle** field in the current LogPen (section [2.2.19](#Section_93ce3f4537ac4affb6e82f6db054c4c4)) and LogPenEx (section [2.2.20](#Section_5b67b3eeea004f809b732959804381be)) objects.  The EMR\_CLOSEFIGURE record SHOULD be used only if there is an open figure in the path bracket. A figure in a path is open unless it is explicitly closed by processing this record. A figure can be open even if the current point is the same as the starting point.  After processing the EMR\_CLOSEFIGURE record, adding a line or curve to the path bracket starts a new figure. |
| EMR\_ENDPATH | This record closes path bracket construction and selects the path into the playback device context. |
| EMR\_FLATTENPATH | This record transforms each curve in the current path into a sequence of lines. |
| EMR\_WIDENPATH | This record redefines the current path as the area that would be painted if its path were drawn using the current pen. |

The generic structure of path bracket records is specified as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that defines the type of the record. The types of records that specify no parameters are listed in the following table. See the preceding table for descriptions of these records.

| Name | Value |
| --- | --- |
| EMR\_BEGINPATH | 0x0000003B |
| EMR\_ENDPATH | 0x0000003C |
| EMR\_CLOSEFIGURE | 0x0000003D |
| EMR\_FLATTENPATH | 0x00000041 |
| EMR\_WIDENPATH | 0x00000042 |
| EMR\_ABORTPATH | 0x00000044 |

**Size (4 bytes):** An unsigned integer that specifies the size in bytes of this record in the [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7). For path bracket records, this value is 0x00000008

See section [2.3](#Section_e0137630f3ad492cbde9e68866e255ba) for more EMF record types.

### State Record Types

The **State** record types specify graphics properties that define the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45) during the processing of [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7) records. For more information about the state used for metafile playback, see the Abstract Data Model section [3.1.1](#Section_fd90d425af0c493dbcb5f7d927f5ca20).

**Note:** The EMR\_REALIZEPALETTE and EMR\_SAVEDC records do not specify parameters.

The following are EMF state record types.

| Name | Section | Description |
| --- | --- | --- |
| EMR\_COLORMATCHTOTARGETW | [2.3.11.1](#Section_b9e3e8fa1d074475b0f2d157eeca5417) | Specifies how to preview colors as they would appear on the target device. |
| EMR\_FORCEUFIMAPPING | [2.3.11.2](#Section_e9a05924e704403783126bef2e19f6bd) | Forces the [**font mapper**](#gt_3c6f410b-e1f3-4841-8254-6f45433425f1) to match fonts based on their **UniversalFontId** in preference to LogFont (section [2.2.13](#Section_1cd7cb7a1a7c48d2b4a4c218f9d12100)). |
| EMR\_INVERTRGN | [2.3.11.3](#Section_5cbc44dead4047b6a0f86179b9eef738) | Inverts the colors in the specified [**region**](#gt_6f13fe5f-8747-4ae0-9375-814bf0528197). |
| EMR\_MOVETOEX | [2.3.11.4](#Section_8ae57e5a22e3435298ca09628f47c693) | Specifies the coordinates of a new drawing position, in logical units. |
| EMR\_PIXELFORMAT | [2.3.11.5](#Section_ba9aa909eb054d2fa51b10cb6df68a54) | Specifies the pixel format. |
| EMR\_REALIZEPALETTE | 2.3.11 | This record maps [**palette**](#gt_dca5cd91-8d01-424b-9722-c9e818562ee5) entries from the current LogPalette object (section [2.2.17](#Section_758f047d765e424c9204a833b7b4e527)) to the system\_palette.  This record specifies no parameters. |
| EMR\_RESTOREDC | [2.3.11.6](#Section_efa2f1804c7e4a7bb4efa1d06a0ce89f) | Restores the playback device context to the specified state, which was saved by a preceding **EMR\_SAVEDC** record. |
| EMR\_SAVEDC | 2.3.11 | Saves the current state of the playback device context in an array of states saved by preceding **EMR\_SAVEDC** records if any.  An **EMR\_RESTOREDC** record is used to restore the state.  This record specifies no parameters. |
| EMR\_SCALEVIEWPORTEXTEX | [2.3.11.7](#Section_469e353c42094919a5aa9331b60765ed) | Specifies the viewport by using the ratios formed by the specified multiplicands and divisors. |
| EMR\_SCALEWINDOWEXTEX | [2.3.11.8](#Section_01698f29bbcb41009bad0fd52c33b0da) | Specifies the window by using the ratios formed by the specified multiplicands and divisors. |
| EMR\_SETARCDIRECTION | [2.3.11.9](#Section_83a955e613b84720a521542e3dd91cba) | Specifies the drawing direction to be used for arc and rectangle output. |
| EMR\_SETBKCOLOR | [2.3.11.10](#Section_24f4095262d440b0ba40aad8fb4fc421) | Specifies the background color. |
| EMR\_SETBKMODE | [2.3.11.11](#Section_7c584a42b6dd47b8a0b6f529133292f4) | Specifies the background mode, which determines how to combine the background with foreground text, hatched brushes, and pen styles that are not solid lines. |
| EMR\_SETBRUSHORGEX | [2.3.11.12](#Section_b02ab7ee4c884bf29ce99100636d86be) | Specifies the origin of the current brush. |
| EMR\_SETCOLORADJUSTMENT | [2.3.11.13](#Section_81b3cd7ced6742aaab9847b99d697231) | Specifies color adjustment values to use in [**bitmap**](#gt_26a17e58-eeba-4210-81b9-8bbadf7ee952) stretching. |
| EMR\_SETICMMODE | [2.3.11.14](#Section_a7c9d4834a2440588dc1b5e0809ea758) | Specifies [**ICM**](#gt_fcd5ee55-57ff-45d6-b51e-2ccba47329e3) to be enabled, disabled, or queried. |
| EMR\_SETICMPROFILEA | [2.3.11.15](#Section_cf1bc41cbdc54645890fc03e05d4d26d) | Specifies how to set a specified [**color profile**](#gt_cffe5dfa-5408-4302-8ab0-400cb94f5e32) as the output profile. |
| EMR\_SETICMPROFILEW | [2.3.11.16](#Section_1d376b73e0084440b6b95770d4afdee5) | Specifies how to set a specified color profile as the output profile. |
| EMR\_SETLAYOUT | [2.3.11.17](#Section_dcb69f9d6f69466aaf1e6f505d86cc67) | Specifies the layout of the playback device context. |
| EMR\_SETLINKEDUFIS | [2.3.11.18](#Section_3bb62157624f443fa88cef5e696740dd) | Sets the **UniversalFontId** ([section 2.2.27](#Section_34502dab22b44855b9b288f00a4a0c52)) of the linked fonts to use during character lookup. |
| EMR\_SETMAPMODE | [2.3.11.19](#Section_aa4ad35dfa424a4f959a8b41304e1b05) | Specifies the [**mapping mode**](#gt_6e998e8d-4394-4263-8632-503730f0abae). |
| EMR\_SETMAPPERFLAGS | [2.3.11.20](#Section_6647abf7bd7043ef87661a6e06ed2f03) | Specifies the algorithm the font mapper uses when it maps logical fonts to physical fonts. |
| EMR\_SETMITERLIMIT | [2.3.11.21](#Section_2637d0d092dd48e8be5e6400b4d1fc5e) | Specifies the limit for the length of miter joins. |
| EMR\_SETPOLYFILLMODE | [2.3.11.22](#Section_e8799c80fe7445c1bf6d3eb78edd4ed2) | Defines polygon fill mode. |
| EMR\_SETROP2 | [2.3.11.23](#Section_cedee7e04e6647b6ae1d816e1e319786) | Defines a binary [**raster operation**](#gt_772681cd-b9a5-440c-8021-b6219c70f6fc) mode. |
| EMR\_SETSTRETCHBLTMODE | [2.3.11.24](#Section_ff00ab986e2c4ecab66404ab2956b6a5) | Specifies bitmap stretch mode. |
| EMR\_SETTEXTALIGN | [2.3.11.25](#Section_a49bb565a58341da8d4a8a0314b3e397) | Specifies text alignment. |
| EMR\_SETTEXTCOLOR | [2.3.11.26](#Section_1ae5c616cf8e4f5da7c3bfc1f70ece28) | Defines the current text color. |
| EMR\_SETTEXTJUSTIFICATION | [2.3.11.27](#Section_edbc2be01da045d69c05677cbcaa2d47) | Sets the amount of extra space to add to break characters for justification purposes. |
| EMR\_SETVIEWPORTEXTEX | [2.3.11.28](#Section_a71486a39ebd43e3819f5318a58b70d5) | Defines the viewport extent. |
| EMR\_SETVIEWPORTORGEX | [2.3.11.29](#Section_b89489473f8d4adda4b8e572e8f90506) | Defines the viewport origin. |
| EMR\_SETWINDOWEXTEX | [2.3.11.30](#Section_1952148360b549b8b58c4b5d1b6cf839) | Defines the window extent. |
| EMR\_SETWINDOWORGEX | [2.3.11.31](#Section_952803a93d5c4c83a258558bc1123bdb) | Defines the window origin. |

The generic structure of state records is specified as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| StateRecordBuffer (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that defines the type of record. The state record types are listed in the following table. See the preceding table for descriptions of these record types.

| Name | Value |
| --- | --- |
| EMR\_SETWINDOWORGEX | 0x0000000A |
| EMR\_SETVIEWPORTEXTEX | 0x0000000B |
| EMR\_SETVIEWPORTORGEX | 0x0000000C |
| EMR\_SETBRUSHORGEX | 0x0000000D |
| EMR\_SETCOLORADJUSTMENT | 0x00000017 |
| EMR\_MOVETOEX | 0x0000001B |
| EMR\_SCALEVIEWPORTEXTEX | 0x0000001F |
| EMR\_SETWINDOWEXTEX | 0x00000009 |
| EMR\_SETMAPPERFLAGS | 0x00000010 |
| EMR\_SETMAPMODE | 0x00000011 |
| EMR\_SETBKMODE | 0x00000012 |
| EMR\_SETPOLYFILLMODE | 0x00000013 |
| EMR\_SETROP2 | 0x00000014 |
| EMR\_SETSTRETCHBLTMODE | 0x00000015 |
| EMR\_SETTEXTALIGN | 0x00000016 |
| EMR\_SETTEXTCOLOR | 0x00000018 |
| EMR\_SETBKCOLOR | 0x00000019 |
| EMR\_SCALEWINDOWEXTEX | 0x00000020 |
| EMR\_SAVEDC | 0x00000021 |
| EMR\_RESTOREDC | 0x00000022 |
| EMR\_REALIZEPALETTE | 0x00000034 |
| EMR\_SETARCDIRECTION | 0x00000039 |
| EMR\_SETMITERLIMIT | 0x0000003A |
| EMR\_INVERTRGN | 0x00000049 |
| EMR\_SETICMMODE | 0x00000062 |
| EMR\_PIXELFORMAT | 0x00000068 |
| EMR\_FORCEUFIMAPPING | 0x0000006D |
| EMR\_SETICMPROFILEA | 0x00000070 |
| EMR\_SETICMPROFILEW | 0x00000071 |
| EMR\_SETLAYOUT | 0x00000073 |
| EMR\_SETLINKEDUFIS | 0x00000077 |
| EMR\_SETTEXTJUSTIFICATION | 0x00000078 |
| EMR\_COLORMATCHTOTARGETW | 0x00000079 |

**Size (4 bytes):** An unsigned integer that specifies the size in bytes of this record in the metafile. This value MUST be a multiple of 4 bytes.

**StateRecordBuffer (variable, optional):** An array of bytes that contains the remainder of the state record.

The EMR\_REALIZEPALETTE and EMR\_SAVEDC records do not contain this field.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| StateRecordParm (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| AlignmentPadding (variable, optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**StateRecordParm (variable):** An array of bytes that contains the parameters for the state record.

**AlignmentPadding (variable, optional):** An array of up to 3 bytes that pads the record so that its total size is a multiple of 4 bytes. This field MUST be ignored.

See section [2.3](#Section_e0137630f3ad492cbde9e68866e255ba) for more [EMF record types](#Section_1eec80ba799b4784a9ac91597d590ae1).

#### EMR\_COLORMATCHTOTARGETW Record

The **EMR\_COLORMATCHTOTARGETW** record specifies whether to perform [**color matching**](#gt_a3322d29-7dc5-4696-b17a-55311db1cb33) with a [**color profile**](#gt_cffe5dfa-5408-4302-8ab0-400cb94f5e32) that is specified in a file with a name consisting of [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) characters.[<81>](#Appendix_A_81" \o "Product behavior note 81)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwAction | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwFlags | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbName | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbData | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as EMR\_COLORMATCHTOTARGETW. This value is 0x00000079.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**dwAction (4 bytes):** An unsigned integer that specifies a value from the ColorSpace enumeration (section [2.1.7](#Section_75da12f8f43748e886774b46e0045969)).

**dwFlags (4 bytes):** An unsigned integer that specifies a value from the ColorMatchToTarget enumeration (section [2.1.6](#Section_b63c02c176444ef3a4963127621cb7b5)).

**cbName (4 bytes):** An unsigned integer that specifies the number of bytes in the Unicode [**UTF16-LE**](#gt_f25550c9-f84f-4eb2-8156-14794a7e3059) name of the target color profile.

**cbData (4 bytes):** An unsigned integer that specifies the size of the raw data of the target color profile in the **Data** field.

**Data (variable):** An array of size (**cbName** + **cbData**) bytes, which specifies the UTF16-LE name and raw data of the target color profile.

An EMR\_COLORMATCHTOTARGETW record can be used to control whether to apply the current color [**transform**](#gt_fdca4a96-8e04-4dc0-acad-1d01e1384a5b) to subsequent graphics operations. If the **dwAction** field value is **CS\_ENABLE**, [**color mapping**](#gt_a216d932-5b97-4882-b1a1-58c47532795c) is enabled, and the current color transform SHOULD be applied. If **dwAction** is set to **CS\_DISABLE**, the color transform SHOULD NOT be applied.

Before applying the current color transform, [**WCS**](#gt_e4f9c4a4-c195-43d0-b189-f9727ec8a2b3) SHOULD be enabled in the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45).[<82>](#Appendix_A_82" \o "Product behavior note 82)

While color mapping to the target is enabled by a **dwAction** value of **CS\_ENABLE**, changes to the [**color space**](#gt_5d3fb5ea-c686-4d3b-b0ff-aef33fe1aee5) or [**color gamut**](#gt_cc6ddf19-b7de-4820-8b48-ecd73c45eef2) mapping are not applied. However, those changes MUST take effect when color mapping to the target is disabled.

The **dwAction** field SHOULD NOT be set to **CS\_DELETE\_TRANSFORM** unless color management has already been enabled with an EMR\_SETICMMODE record (section [2.3.11.14](#Section_a7c9d4834a2440588dc1b5e0809ea758)).

See section [2.3.11](#Section_8f8a4df2f8d74d39afc094e19f524652) for more state record types.

#### EMR\_FORCEUFIMAPPING Record

The **EMR\_FORCEUFIMAPPING** record forces the font mapper to match fonts based on their **UniversalFontId** in preference to their LogFont (section [2.2.13](#Section_1cd7cb7a1a7c48d2b4a4c218f9d12100)) information.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ufi | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as EMR\_FORCEUFIMAPPING. This value is 0x0000006D.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**ufi (8 bytes):** The font id to use, specified as a UniversalFontId ([section 2.2.27](#Section_34502dab22b44855b9b288f00a4a0c52)).

See section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4) for more drawing record types.

#### EMR\_INVERTRGN Record

The **EMR\_INVERTRGN** record inverts the colors in the specified [**region**](#gt_6f13fe5f-8747-4ae0-9375-814bf0528197). The current clipping regions used by this record are maintained in a **Regions** state element (section [3.1.1.2.1](#Section_939400bc0ecd4a28aa8c60c7e4cc9877)) in the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45) (section [3.1](#Section_50ae16a09d1748018bf1e28e9533a889)).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 |  | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| RgnDataSize | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| RgnData (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as EMR\_INVERTRGN. This value is 0x00000049.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Bounds (16 bytes):** A RectL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.19) that specifies the destination bounding rectangle in logical coordinates. If the intersection of this rectangle with the current clipping region is empty, this record has no effect.

**RgnDataSize (4 bytes):** An unsigned integer that specifies the size of region data in bytes.

**RgnData (variable):** A **RgnDataSize** length array of bytes that specifies the output region in a **RegionData** object (section [2.2.24](#Section_e66601f29b5c46198476ddb7b087551b)). The bounds specified by the **RegionDataHeader** field of this object MAY[<83>](#Appendix_A_83" \o "Product behavior note 83) be used as the bounding rectangle of the region when this record is processed.

If the output surface is monochrome, this record SHOULD convert white pixels to black and black pixels to white. For color output, the inversion is dependent on the type of technology used to generate the colors.

See section [2.3.11](#Section_8f8a4df2f8d74d39afc094e19f524652) for more state record types.

#### EMR\_MOVETOEX Record

The EMR\_MOVETOEX record specifies the coordinates of s new drawing position in logical units.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Offset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_MOVETOEX**. This value is 0x0000001B.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Offset (8 bytes):** A PointL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.15), which specifies coordinates of the new drawing position in logical units.

See section [2.3.11](#Section_8f8a4df2f8d74d39afc094e19f524652) for more state record types.

#### EMR\_PIXELFORMAT Record

The EMR\_PIXELFORMAT record specifies the pixel format to use for graphics operations.[<84>](#Appendix_A_84" \o "Product behavior note 84)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| pfd (40 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_PIXELFORMAT**. This value is 0x00000068.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**pfd (40 bytes):** A PixelFormatDescriptor object (section [2.2.22](#Section_1db036d62da84b92b4f8e9cab8cc93b7)) that specifies pixel format data.

See section [2.3.11](#Section_8f8a4df2f8d74d39afc094e19f524652) for more state record types.

#### EMR\_RESTOREDC Record

The **EMR\_RESTOREDC** record restores the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45) to the specified state. The playback device context is restored by popping state information off a stack that was created by a prior EMR\_SAVEDCrecord ([section 2.3.11](#Section_8f8a4df2f8d74d39afc094e19f524652)).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SavedDC | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies the record type as **EMR\_RESTOREDC**. This value is 0x00000022.

**Size (4 bytes):** An unsigned integer that specifies the size of the record in bytes. This value is 0x0000000C.

**SavedDC (4 bytes):** A signed integer that specifies the saved state to restore relative to the current state. This value MUST be negative; –1 represents the state that was most recently saved on the stack, –2 the one before that, etc.

The stack can contain state information for multiple instances of the playback device context. When a state is restored, all state instances that were saved more recently MUST be discarded.

See section 2.3.11 for more state record types.

#### EMR\_SCALEVIEWPORTEXTEX Record

The **EMR\_SCALEVIEWPORTEXTEX** record specifies the current viewport in the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45) by using ratios formed by the specified multiplicands and divisors.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| xNum | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| xDenom | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| yNum | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| yDenom | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as EMR\_SCALEVIEWPORTEXTEX. This value is 0x0000001F.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**xNum (4 bytes):** A signed integer that specifies the horizontal multiplicand. Cannot be zero.

**xDenom (4 bytes):** A signed integer that specifies the horizontal divisor. Cannot be zero.

**yNum (4 bytes):** A signed integer that specifies the vertical multiplicand. Cannot be zero.

**yDenom (4 bytes):** A signed integer that specifies the vertical divisor. Cannot be zero.

The extent MUST NOT be changed if the current [**mapping mode**](#gt_6e998e8d-4394-4263-8632-503730f0abae) (section [2.1.21](#Section_109d7955767e4955a0e2f8b5e14f757a)) is fixed scale. Only MM\_ISOTROPIC and MM\_ANISOTROPIC are not fixed scale.

The new viewport extent is computed as follows.

1. xNewWE = (xOldWE \* xNum) / xDenom
2. yNewWE = (yOldWE \* yNum) / yDenom

See section [2.3.11](#Section_8f8a4df2f8d74d39afc094e19f524652) for more state record types.

#### EMR\_SCALEWINDOWEXTEX Record

The **EMR\_SCALEWINDOWEXTEX** record specifies the current window in the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45) by using ratios formed by the specified multiplicands and divisors.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| xNum | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| xDenom | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| yNum | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| yDenom | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as EMR\_SCALEWINDOWEXTEX. This value is 0x00000020.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**xNum (4 bytes):** A signed integer that specifies the horizontal multiplicand. MUST NOT be zero.

**xDenom (4 bytes):** A signed integer that specifies the horizontal divisor. MUST NOT be zero.

**yNum (4 bytes):** A signed integer that specifies the vertical multiplicand. MUST NOT be zero.

**yDenom (4 bytes):** A signed integer that specifies the vertical divisor. MUST NOT be zero.

The extent MUST NOT be changed if the current [**mapping mode**](#gt_6e998e8d-4394-4263-8632-503730f0abae) (section [2.1.21](#Section_109d7955767e4955a0e2f8b5e14f757a)) is fixed scale. Only MM\_ISOTROPIC and MM\_ANISOTROPIC are not fixed scale.

The new window extent is computed as follows.

1. xNewWE = (xOldWE \* xNum) / xDenom
2. yNewWE = (yOldWE \* yNum) / yDenom

See section [2.3.11](#Section_8f8a4df2f8d74d39afc094e19f524652) for more state record types.

#### EMR\_SETARCDIRECTION Record

The **EMR\_SETARCDIRECTION** record specifies the drawing direction to be used for arc and rectangle output.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ArcDirection | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as EMR\_SETARCDIRECTION. This value is 0x00000039.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes. This value is 0x0000000C.

**ArcDirection (4 bytes):** An unsigned integer that specifies the arc direction. This value is in the ArcDirection enumeration (section [2.1.2](#Section_9886c62f17884118894a76d5a8733685)). The default direction is counterclockwise.

The arc direction affects the direction in which the following records draw:

* [EMR\_ARC (section 2.3.5.2)](#Section_3e148f197e7543aa9259fda562f60315)
* [EMR\_ARCTO (section 2.3.5.3)](#Section_51b4a890bfd4496ebbd9dbde9ad49d1c)
* [EMR\_CHORD (section 2.3.5.4)](#Section_b442d2fbf92b4103aa7d44f864f719d5)
* [EMR\_ELLIPSE (section 2.3.5.5)](#Section_d9462a32d18840e7bf7205614ec4ff18)
* [EMR\_PIE (section 2.3.5.15)](#Section_c326f6b2ae7e447fb8dc02c703f054c3)
* [EMR\_RECTANGLE (section 2.3.5.34)](#Section_3c4712380a02499290a2bfd2afd98f2a)
* [EMR\_ROUNDRECT (section 2.3.5.35)](#Section_2e4f55a15f694a818329423eeedb3812)

See section [2.3.11](#Section_8f8a4df2f8d74d39afc094e19f524652) for more state record types.

#### EMR\_SETBKCOLOR Record

The **EMR\_SETBKCOLOR** record specifies the background color for text output.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Color | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as EMR\_SETBKCOLOR. This value is 0x00000019.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Color (4 bytes):** A ColorRef object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.8), which specifies the background color value.

See section [2.3.11](#Section_8f8a4df2f8d74d39afc094e19f524652) for more state record types.

#### EMR\_SETBKMODE Record

The **EMR\_SETBKMODE** record specifies the background mix mode to use with text, hatched brushes, and pens that are not solid lines.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BackgroundMode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as EMR\_SETBKMODE. This value is 0x00000012.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**BackgroundMode (4 bytes):** An unsigned integer that specifies the background mode, from the BackgroundMode enumeration (section [2.1.4](#Section_2f57054c2b694303a6f85aacd401f759)).

See section [2.3.11](#Section_8f8a4df2f8d74d39afc094e19f524652) for more state record types.

#### EMR\_SETBRUSHORGEX Record

The **EMR\_SETBRUSHORGEX** record specifies the origin of the current brush.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Origin | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as EMR\_SETBRUSHORGEX. This value is 0x0000000D.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Origin (8 bytes):** A PointL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.15), which specifies the horizontal and vertical origin of the current brush in logical units.

See section [2.3.11](#Section_8f8a4df2f8d74d39afc094e19f524652) for more state record types.

#### EMR\_SETCOLORADJUSTMENT Record

The **EMR\_SETCOLORADJUSTMENT** record specifies color adjustment properties in the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ColorAdjustment (24 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_SETCOLORADJUSTMENT**. This value is 0x00000017.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes. This value is 0x00000020.

**ColorAdjustment (24 bytes):** A ColorAdjustment object (section [2.2.2](#Section_71d7f4d0b6454de8a5588a2e0fc596cb)) that specifies color adjustment values.

Color adjustment values are used to adjust the input color of the source [**bitmap**](#gt_26a17e58-eeba-4210-81b9-8bbadf7ee952) for graphics operations performed by [EMR\_STRETCHBLT](#Section_0dd551f380f7485289c29ddba803a192) and [EMR\_STRETCHDIBITS](#Section_89c0d8080dea413fbe402e9e51fa36ac) records when **STRETCH\_HALFTONE** mode is set from the [StretchMode](#Section_e72f20aa2e2d453f9e95eaef50b76e19) enumeration (section 2.1.32).

The ColorAdjustment object specified by this record MUST be used in graphics operations that require a ColorAdjustment object, until a different ColorAdjustment object is specified by another EMR\_SETCOLORADJUSTMENT record, or until the object is removed by a [EMR\_DELETEOBJECT](#Section_6f0f12a3111a478b8251a9505168f9a9) record.

See section [2.3.11](#Section_8f8a4df2f8d74d39afc094e19f524652) more state record types.

#### EMR\_SETICMMODE Record

The **EMR\_SETICMMODE** record specifies the mode of [**Image Color Management (ICM)**](#gt_fcd5ee55-57ff-45d6-b51e-2ccba47329e3) for graphics operations.[<85>](#Appendix_A_85" \o "Product behavior note 85)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ICMMode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_SETICMMODE**. This value is 0x00000062.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes. This value is 0x0000000C.

**ICMMode (4 bytes):** An unsigned integer that specifies whether to enable or disable ICM, from the ICMMode enumeration (section [2.1.18](#Section_4935fb3cc6464d16a8f0f86ee0a9d096)).

When ICM mode is enabled in the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45), colors specified in EMF records SHOULD be [**color matched**](#gt_a3322d29-7dc5-4696-b17a-55311db1cb33), whereas the default [**color profile**](#gt_cffe5dfa-5408-4302-8ab0-400cb94f5e32) SHOULD be used when a bit-block transfer is performed. If the default color profile is not desired, ICM mode SHOULD be turned off before performing the bit-block transfer.

See section [2.3.11](#Section_8f8a4df2f8d74d39afc094e19f524652) for more state record types.

#### EMR\_SETICMPROFILEA Record

The **EMR\_SETICMPROFILEA** record specifies a [**color profile**](#gt_cffe5dfa-5408-4302-8ab0-400cb94f5e32) in a file with a name consisting of [**ASCII**](#gt_79fa85ca-ac61-467c-b819-e97dc1a7a599) characters, for graphics output.[<86>](#Appendix_A_86" \o "Product behavior note 86)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwFlags | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbName | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbData | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as EMR\_SETICMPROFILEA. This value is 0x00000070.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**dwFlags (4 bytes):** An unsigned integer that contains color profile flags.

**cbName (4 bytes):** An unsigned integer that specifies the number of bytes in the ASCII name of the desired color profile.

**cbData (4 bytes):** An unsigned integer that specifies the size of the color profile data, if it is contained in the **Data** field.

**Data (variable):** An array of size (**cbName** + **cbData**) in bytes, which specifies the ASCII name and raw data of the desired color profile.

See section [2.3.11](#Section_8f8a4df2f8d74d39afc094e19f524652) for more state record types.

#### EMR\_SETICMPROFILEW Record

The **EMR\_SETICMPROFILEW** record specifies a [**color profile**](#gt_cffe5dfa-5408-4302-8ab0-400cb94f5e32) in a file with a name consisting of [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) characters, for graphics output.[<87>](#Appendix_A_87" \o "Product behavior note 87)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwFlags | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbName | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbData | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_SETICMPROFILEW**. This value is 0x00000071.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**dwFlags (4 bytes):** An unsigned integer that contains color profile flags.

**cbName (4 bytes):** An unsigned integer that specifies the number of bytes in the Unicode [**UTF16-LE**](#gt_f25550c9-f84f-4eb2-8156-14794a7e3059) name of the desired color profile.

**cbData (4 bytes):** An unsigned integer that specifies the size of color profile data, if attached.

**Data (variable):** An array of size (**cbName** + **cbData**) in bytes, which specifies the UTF16-LE name and raw data of the desired color profile.

See section [2.3.11](#Section_8f8a4df2f8d74d39afc094e19f524652) for more state record types.

#### EMR\_SETLAYOUT Record

The **EMR\_SETLAYOUT** record specifies the order in which text and graphics are drawn.[<88>](#Appendix_A_88" \o "Product behavior note 88)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| LayoutMode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as EMR\_SETLAYOUT. This value is 0x00000073.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes. This value is 0x0000000C.

**LayoutMode (4 bytes):** An unsigned integer that specifies the layout mode as follows:

| Value | Meaning |
| --- | --- |
| LAYOUT\_LTR  0x00000000 | Sets the default horizontal layout to be left-to-right. This is the default mode for English and European locales. |
| LAYOUT\_RTL  0x00000001 | Sets the default horizontal layout to be right-to-left. This mode is required for some languages, including Arabic and Hebrew. |
| LAYOUT\_BITMAPORIENTATIONPRESERVED  0x00000008 | Disables mirroring of bitmaps that are drawn by bitmap records (section [2.3.1](#Section_2a57d4bdf3dc45df95fd447382b5262a)) when the layout mode is right-to-left. |

See section [2.3.11](#Section_8f8a4df2f8d74d39afc094e19f524652) for more state record types.

#### EMR\_SETLINKEDUFIS Record

The **EMR\_SETLINKEDUFIS** record sets the **UniversalFontIds** ([section 2.2.27](#Section_34502dab22b44855b9b288f00a4a0c52)) of linked fonts to use during character lookup.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| uNumLinkedUFI | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ufis (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_SETLINKEDUFIS**. This value is 0x00000077.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**uNumLinkedUFI (4 bytes):** An unsigned integer specifying the number of UFIs to follow.

**ufis (variable):** An array of **uNumLinkedUFI** elements of type UniversalFontId (section 2.2.27), which specifies the identifiers of the linked fonts.

**Reserved (8 bytes):** This field is reserved and MUST be ignored.

See section [2.3.11](#Section_8f8a4df2f8d74d39afc094e19f524652) for more state record types.

#### EMR\_SETMAPMODE Record

The **EMR\_SETMAPMODE** record specifies the current [**mapping mode**](#gt_6e998e8d-4394-4263-8632-503730f0abae), which specifies the unit of measure used to transform [**page space**](#gt_f530f123-858a-4af7-b626-adea0aeaa75a) units into [**device space**](#gt_25d586c7-f440-45c6-b965-5c025bea9fc0) units, and also specifies the orientation of the device's x-axis and y-axis.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MapMode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_SETMAPMODE**. This value is 0x00000011.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**MapMode (4 bytes):** An unsigned integer from the MapMode enumeration (section [2.1.21](#Section_109d7955767e4955a0e2f8b5e14f757a)).

**MM\_TEXT** mode allows applications to work in device pixels, whose size varies from device to device.

The **MM\_HIENGLISH**, **MM\_HIMETRIC**, **MM\_LOENGLISH**, **MM\_LOMETRIC**, and **MM\_TWIPS** modes are useful for applications drawing in physically meaningful units such as inches or millimeters.

**MM\_ISOTROPIC** mode ensures a 1:1 aspect ratio.

**MM\_ANISOTROPIC** mode allows the x-coordinates and y-coordinates to be adjusted independently.

See section [2.3.11](#Section_8f8a4df2f8d74d39afc094e19f524652) for more state record types.

#### EMR\_SETMAPPERFLAGS Record

The **EMR\_SETMAPPERFLAGS** record specifies parameters for the process of matching logical fonts to physical fonts, which is performed by the [**font mapper**](#gt_3c6f410b-e1f3-4841-8254-6f45433425f1).[<89>](#Appendix_A_89" \o "Product behavior note 89)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Flags | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as EMR\_SETMAPPERFLAGS. This value is 0x00000010.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes. This value is 0x0000000C.

**Flags (4 bytes):** An unsigned integer that specifies parameters for the font matching process.

| Value | Meaning |
| --- | --- |
| 0x00000000 | The font mapper is not limited to fonts that match the [**aspect ratio**](#gt_091ff8ed-6ac1-4015-9d91-93020f13f6b5) of the output device. |
| 0x00000001 | The font mapper SHOULD select only fonts that match the aspect ratio of the output device. |

See section [2.3.11](#Section_8f8a4df2f8d74d39afc094e19f524652) for more state record types.

#### EMR\_SETMITERLIMIT Record

The EMR\_SETMITERLIMIT record specifies the limit for the length of miter joins.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MiterLimit | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_SETMITERLIMIT**. This value is 0x0000003A.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**MiterLimit (4 bytes):** An unsigned integer that specifies the new [**miter length**](#gt_7d78ab9f-9531-4993-b140-8c2bb6e82220) limit.[<90>](#Appendix_A_90" \o "Product behavior note 90)

See section [2.3.11](#Section_8f8a4df2f8d74d39afc094e19f524652) for more state record types.

#### EMR\_SETPOLYFILLMODE Record

The **EMR\_SETPOLYFILLMODE** record defines polygon fill mode.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PolygonFillMode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as EMR\_SETPOLYFILLMODE. This value is 0x00000013.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**PolygonFillMode (4 bytes):** An unsigned integer that specifies the polygon fill mode and is in the PolygonFillMode (section [2.1.27](#Section_879a186908ad497783e809be265e2c7c)) enumeration.

In general, the modes differ only in cases where a complex, overlapping polygon MUST be filled; for example, a five-sided polygon that forms a five-pointed star with a pentagon in the center. In such cases, ALTERNATE mode SHOULD fill every other enclosed [**region**](#gt_6f13fe5f-8747-4ae0-9375-814bf0528197) within the polygon (the points of the star), but WINDING mode SHOULD fill all regions (the points of the star and the pentagon).

When the fill mode is ALTERNATE, the area between odd-numbered and even-numbered polygon sides on each scan line SHOULD be filled. That is, the area between the first and second side SHOULD be filled, and between the third and fourth side, and so on.

When the fill mode is WINDING, any region that has a nonzero winding value SHOULD be filled. The winding value is the number of times a pen used to draw the polygon would go around the region. The direction of each edge of the polygon is significant.

See section [2.3.11](#Section_8f8a4df2f8d74d39afc094e19f524652) for more state record types.

#### EMR\_SETROP2 Record

The EMR\_SETROP2 record defines a binary [**raster operation**](#gt_772681cd-b9a5-440c-8021-b6219c70f6fc) mode.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ROP2Mode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_SETROP2**. This value is 0x00000014.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**ROP2Mode (4 bytes):** An unsigned integer that specifies the raster operation mode and is in the Binary Raster Op enumeration ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.1.1.2).

Binary raster operation mix modes define how to combine source and destination colors when drawing with the current pen. The mix modes are binary raster operation codes, representing all possible [**Boolean**](#gt_1d79d7a7-ba2c-4b34-931c-7ba8057c87b2) functions of two variables, using the binary operations AND, OR, and XOR (exclusive OR), and the unary operation NOT. The mix mode is for raster devices only; it is not available for vector devices.

See section [2.3.11](#Section_8f8a4df2f8d74d39afc094e19f524652) for more state record types.

#### EMR\_SETSTRETCHBLTMODE Record

The EMR\_SETSTRETCHBLTMODE record specifies [**bitmap**](#gt_26a17e58-eeba-4210-81b9-8bbadf7ee952) stretch mode.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| StretchMode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_SETSTRETCHBLTMODE**. This value is 0x00000015.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**StretchMode (4 bytes):** An unsigned integer that specifies the stretch mode and MAY be in the [StretchMode](#Section_e72f20aa2e2d453f9e95eaef50b76e19) enumeration.

The stretching mode specifies how to combine rows or columns of a bitmap with existing pixels on the display device that the [EMR\_STRETCHBLT](#Section_0dd551f380f7485289c29ddba803a192) record is processed on.

The **STRETCH\_ANDSCANS** and **STRETCH\_ORSCANS** modes are typically used to preserve foreground pixels in monochrome bitmaps. The **STRETCH\_DELETESCANS** mode is typically used to preserve color in color bitmaps.

The **STRETCH\_HALFTONE** mode is slower and requires more processing of the source image than the other three modes, but produces higher quality images. Also note that an [EMR\_SETBRUSHORGEX](#Section_b02ab7ee4c884bf29ce99100636d86be) SHOULD be encountered after setting the **STRETCH\_HALFTONE** mode to avoid brush misalignment.

See section [2.3.11](#Section_8f8a4df2f8d74d39afc094e19f524652) for more state record types.

#### EMR\_SETTEXTALIGN Record

The **EMR\_SETTEXTALIGN** record specifies text alignment for text drawing.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| TextAlignmentMode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as EMR\_SETTEXTALIGN. This value is 0x00000016.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**TextAlignmentMode (4 bytes):** An unsigned integer that specifies text alignment by using a mask of text alignment flags. These are either TextAlignmentMode flags ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.1.2.3) for text with a horizontal baseline, or VerticalTextAlignmentMode flags ([MS-WMF] section 2.1.2.4) for text with a vertical baseline. Only one value can be chosen from those that affect horizontal and vertical alignment.

The EMR\_SMALLTEXTOUT, EMR\_EXTTEXTOUTA, and EMR\_EXTTEXTOUTW records (section [2.3.5](#Section_1f7da2e4df0441d48f7d180fa642cab4)) use text alignment values to position a string of text on the output medium. The values specify the relationship between a reference point and a rectangle that bounds the text. The reference point is either the current drawing position or a point passed to a text output record.

The rectangle that bounds the text is formed by the character cells in the text string.

See section [2.3.11](#Section_8f8a4df2f8d74d39afc094e19f524652) for more state record types.

#### EMR\_SETTEXTCOLOR Record

The **EMR\_SETTEXTCOLOR** record defines the current text foreground color.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Color | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as EMR\_SETTEXTCOLOR. This value is 0x00000018.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Color (4 bytes):** A ColorRef object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.8) that specifies the text foreground color.

See section [2.3.11](#Section_8f8a4df2f8d74d39afc094e19f524652) for more state record types.

#### EMR\_SETTEXTJUSTIFICATION Record

The **EMR\_SETTEXTJUSTIFICATION** record specifies the amount of extra space to add to break characters for text justification.[<91>](#Appendix_A_91" \o "Product behavior note 91)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| nBreakExtra | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| nBreakCount | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as EMR\_SETTEXTJUSTIFICATION. This value is 0x00000078.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**nBreakExtra (4 bytes):** A signed integer that specifies the total amount of extra space to add in logical units.

**nBreakCount (4 bytes):** A signed integer that specifies the number of break characters.

Instead of using this record, an implementation SHOULD use EMR\_EXTTEXTOUTW (section [2.3.5.8](#Section_a59a79ac328e492da34de02727af6edf)) to perform this function.

See section [2.3.11](#Section_8f8a4df2f8d74d39afc094e19f524652) for more state record types.

#### EMR\_SETVIEWPORTEXTEX Record

The **EMR\_SETVIEWPORTEXTEX** record defines the viewport extent.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Extent | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as EMR\_SETVIEWPORTEXTEX. This value is 0x0000000B.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Extent (8 bytes):** A SizeL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.22) that specifies the horizontal and vertical extents in device units.

See section [2.3.11](#Section_8f8a4df2f8d74d39afc094e19f524652) for more state record types.

#### EMR\_SETVIEWPORTORGEX Record

The **EMR\_SETVIEWPORTORGEX** record defines the viewport origin.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Origin | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as EMR\_SETVIEWPORTORGEX. This value is 0x0000000C.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Origin (8 bytes):** A PointL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.15) that specifies the window horizontal and vertical origin in device units.

See section [2.3.11](#Section_8f8a4df2f8d74d39afc094e19f524652) for more state record types.

#### EMR\_SETWINDOWEXTEX Record

The **EMR\_SETWINDOWEXTEX** record defines the window extent.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Extent | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as EMR\_SETWINDOWEXTEX. This value is 0x00000009.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Extent (8 bytes):** A SizeL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.22) that specifies the horizontal and vertical extents in logical units.

See section [2.3.11](#Section_8f8a4df2f8d74d39afc094e19f524652) for more state record types.

#### EMR\_SETWINDOWORGEX Record

The **EMR\_SETWINDOWORGEX** record defines the window origin.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Origin | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as EMR\_SETWINDOWORGEX. This value is 0x0000000A.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes.

**Origin (8 bytes):** A PointL object ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.15) that specifies the window horizontal and vertical origin in logical units.

See section [2.3.11](#Section_8f8a4df2f8d74d39afc094e19f524652) for more state record types.

### Transform Record Types

The transform record types specify and modify [**world-space**](#gt_baadb707-1343-413b-9b36-2496b06e9bd7) to [**page-space**](#gt_f530f123-858a-4af7-b626-adea0aeaa75a) [**transforms**](#gt_fdca4a96-8e04-4dc0-acad-1d01e1384a5b).

The following are [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) transform record types.

| Name | Section | Description |
| --- | --- | --- |
| EMR\_MODIFYWORLDTRANSFORM | [2.3.12.1](#Section_c70b85e58c31418fa7b8349e417e0f76) | Modifies the current world-space to page-space transform. |
| EMR\_SETWORLDTRANSFORM | [2.3.12.2](#Section_985724c04db148f0b34667288b3288cb) | Specifies a two-dimensional linear transform between world space and page space. |

The generic structure of EMF transform records is specified as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Xform (24 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| TransformData (optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that defines the type of record. The transform record types are listed in the following table. See the preceding table for descriptions of these record types.

| Name | Value |
| --- | --- |
| EMR\_SETWORLDTRANSFORM | 0x00000023 |
| EMR\_MODIFYWORLDTRANSFORM | 0x00000024 |

**Size (4 bytes):** An unsigned integer that specifies the size in bytes of this record in the [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7). This value MUST be a multiple of 4 bytes.

**Xform (24 bytes):** An [XForm object (section 2.2.28)](#Section_e84107e9bc2b4a1492345d173adc1b59), which defines a world-space to page-space transform.

**TransformData (4 bytes, optional):** An unsigned integer that specifies an additional parameter for the record.

See section [2.3](#Section_e0137630f3ad492cbde9e68866e255ba) for more [EMF record types](#Section_1eec80ba799b4784a9ac91597d590ae1).

#### EMR\_MODIFYWORLDTRANSFORM Record

The EMR\_MODIFYWORLDTRANSFORM record modifies the current [**world-space**](#gt_baadb707-1343-413b-9b36-2496b06e9bd7) to [**page-space**](#gt_f530f123-858a-4af7-b626-adea0aeaa75a) [**transform**](#gt_fdca4a96-8e04-4dc0-acad-1d01e1384a5b) in the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Xform (24 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ModifyWorldTransformMode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_MODIFYWORLDTRANSFORM**. This value is 0x00000024.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes. This value is 0x00000024.

**Xform (24 bytes):** An [XForm object (section 2.2.28)](#Section_e84107e9bc2b4a1492345d173adc1b59) that defines a two-dimensional linear transform in logical units. This transform is used according to the **ModifyWorldTransformMode** to define a new value for the world-space to page-space transform in the playback device context.

**ModifyWorldTransformMode (4 bytes):** An unsigned integer that specifies how the transform specified in **Xform** is used. This value is in the [ModifyWorldTransformMode enumeration (section 2.1.24)](#Section_e6bb2996195f473f80c69dc1afe474f9).

For more information concerning transforms and [**coordinate spaces**](#gt_84184077-d0d9-41e0-acaf-6596aa066813), see [[MSDN-WRLDPGSPC]](https://go.microsoft.com/fwlink/?LinkId=90169). See section [2.3.12](#Section_a0442ad12d6e43e7ac8ce252e492f97e) for more transform record types.

#### EMR\_SETWORLDTRANSFORM Record

The EMR\_SETWORLDTRANSFORM record specifies a [**transform**](#gt_fdca4a96-8e04-4dc0-acad-1d01e1384a5b) for the current [**world-space**](#gt_baadb707-1343-413b-9b36-2496b06e9bd7) to [**page-space**](#gt_f530f123-858a-4af7-b626-adea0aeaa75a) transform in the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Xform (24 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes):** An unsigned integer that identifies this record type as **EMR\_SETWORLDTRANSFORM**. This value is 0x00000023.

**Size (4 bytes):** An unsigned integer that specifies the size of this record in bytes. This value is 0x00000020.

**Xform (24 bytes):** An XForm object (section [2.2.28](#Section_e84107e9bc2b4a1492345d173adc1b59)) that specifies a two-dimensional linear transform in logical units. This transform defines a new value for the current world-space to page-space transform.

For more information concerning transforms and [**coordinate spaces**](#gt_84184077-d0d9-41e0-acaf-6596aa066813), see [[MSDN-WRLDPGSPC]](https://go.microsoft.com/fwlink/?LinkId=90169). See section [2.3.12](#Section_a0442ad12d6e43e7ac8ce252e492f97e) for more transform record types.

# Structure Examples

## EMF Metafile Playback

This section describes the data and processing required for rendering the image stored in an [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7). This process is referred to as “playback” throughout this specification. During metafile playback, the graphics state is maintained in the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45) (section 3.1).

### Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains during the processing of this file format. The description of the organization is provided to facilitate the explanation of that processing. This document does not mandate that implementations adhere to this model, provided their external behavior is consistent with that specified in this document.

The following abstract data model elements are used to maintain the current state of the playback of an [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7).

**EMF object table (variable):** A table of object metadata that is used to keep track of objects created and used during metafile playback. The **EMF object table** structure is described in section [3.1.1.1](#Section_e4fa4e6390964cdcb77685e2a1e4e1f4).

**Graphics Environment (variable):** The set of drawing parameters and metadata that specify the rendering of the image in the metafile. The **Graphics Environment** is described in section [3.1.1.2](#Section_00de3e5e6f104be4943d8de5e2b95fe8).

No part of the abstract data model is assumed to persist across system restarts; however, with this data it would be possible to recover the playback device context and resume metafile playback.

#### EMF Object Table

The **EMF object table** is an element of the state maintained during [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7) playback. It contains data used for managing graphics objects as they are created, activated, used, deactivated, and deleted by the processing of EMF records.

A possible implementation is described by the following elements.

| Element name | Type | Set by |
| --- | --- | --- |
| **EMFObjectArray** | **EMFObject[]** | Object creation record (section [2.3.7](#Section_7d914464d5464393bddb6d1e747fca18)) |
| **EMFObject** | **EMFObject** | Object creation record |

**EMFObjectArray (variable):** An array of **EMFObject** elements. The maximum number of objects is specified in the **Handles** field of the Header object (section [2.2.9](#Section_e4a35c41e8e343f9bc07a18e99bb866d)) in an EMF\_HEADER record (section [2.3.4.2](#Section_de081cd7351f4cc2830bd03fb55e89ab)). The table should be large enough for (**Handles** + 1) because element zero in the array is reserved.

**EMFObject (variable):** A graphics object and its associated index.

When a graphics object is created by an object creation record, the record specifies a numerical index. The object can be referenced by its index during metafile processing until the object is deleted. Object indexes start at 1; zero is reserved for references to the metafile itself.

An object manipulation record (section [2.3.8](#Section_2f1c3268b43b4c28988b41aa49354998)) can use the index of a graphics object to select it into the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45). This has the effect of activating the object so that it can be used in graphics operations specified by subsequent metafile records. Until the object is activated, it is not used. Later, if a different object of the same type is activated, the former object is deactivated but not deleted. An object is not deleted until an object manipulation record is processed that deletes it.

Before a graphics object is instantiated and activated, a default [**stock object**](#gt_b1328cb9-acd8-47f8-9896-fe985af2d6d6) (section [2.1.31](#Section_d6dffd25861542f8aed1309f1fe54ab2)) for that type is used in graphics operations.

| Element name | Type | Set by |
| --- | --- | --- |
| **GraphicsObjectIndex** | UINT32 | Object creation record |
| **GraphicsObject** | See element description. | Object creation record |

**GraphicsObjectIndex (4 bytes):** An unsigned integer that specifies the index associated with the **GraphicsObject**, which is used to refer to the object after it is created.

**GraphicsObject (variable):** One of the following structures.

* DeviceIndependentBitmap ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.9)
* LogBrushEx (section [2.2.12](#Section_6d790f8d51654746bac33443915c2071))
* LogColorSpace ([MS-WMF] section 2.2.2.11)
* LogColorSpaceW ([MS-WMF] section 2.2.2.12)
* LogFont (section [2.2.13](#Section_1cd7cb7a1a7c48d2b4a4c218f9d12100))
* LogFontExDv (section [2.2.15](#Section_595fcaaebc3741489afe859e81ca7f76))
* LogPalette (section [2.2.17](#Section_758f047d765e424c9204a833b7b4e527))
* LogPenEx (section [2.2.20](#Section_5b67b3eeea004f809b732959804381be))

The following process can be used to manage graphics objects with the **EMF object table**:

1. At the start of metafile processing, the **EMFObjectArray** structure is created.
2. The **EMF object table** needs to be large enough to keep track of objects that are explicitly created as well as stock objects. Each element in the object table contains information that indicates whether an object with that index has been created, a way to access the object, and whether the object is currently active.
3. When a graphics object is created, the element in the **EMF object table** that corresponds to its index is updated so that it can be accessed later.
4. When a graphics object is activated, the element in the **EMF object table** that corresponds to its index is updated. In addition, the element that was activated before is now deactivated.
5. When a graphics object is deactivated, the element in the **EMF object table** that corresponds to its index is updated. In addition, the default stock object of that type is now activated.
6. When a record is encountered that deletes the graphics object, its memory is released, and the EMF object table is updated accordingly. A graphics object can be deleted without first being deactivated. If that happens, the default stock object of that type is now activated.

There are some index values that are reserved:

* The index zero is reserved; it refers to the EMF metafile itself.
* Indexes that have the most-significant bit set refer to stock objects.

The object state changes of creation, activation, deactivation, and deletion, require management during playback to achieve the expected results in rendering the image stored in the metafile.

#### Graphics Environment

The **Graphics Environment** describes the graphics state maintained during [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7) playback. A possible implementation is described by the following elements.

| Element name | Type | Set by |
| --- | --- | --- |
| **PlaybackStateArray** | **PlaybackState[]** | EMR\_SAVEDC (section [2.3.11](#Section_8f8a4df2f8d74d39afc094e19f524652)) |
| **PlaybackState** | UINT8[] | EMF records (section [2.3](#Section_e0137630f3ad492cbde9e68866e255ba)) |

**PlaybackStateArray:** An array of saved **PlaybackState** elements, any of which can be used by the EMR\_RESTOREDC record (section [2.3.11.6](#Section_efa2f1804c7e4a7bb4efa1d06a0ce89f)) to restore a previous graphics environment.

**PlaybackState:** The [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45) at some point in EMF record processing, including region definitions, color profiles, fonts and text properties, and graphics drawing metadata. The elements of the **PlaybackState** are grouped as shown in the following table.

| Element group | Section | Description |
| --- | --- | --- |
| **Regions** | [3.1.1.2.1](#Section_939400bc0ecd4a28aa8c60c7e4cc9877) | The current clipping regions and related properties |
| **Colors** | [3.1.1.2.2](#Section_fc99138f11fa48a2903fd6bf332fa9e7) | The current [**color profile**](#gt_cffe5dfa-5408-4302-8ab0-400cb94f5e32) and related properties |
| **Text** | [3.1.1.2.3](#Section_fcf144823aa7424989f6f04dacaf6388) | Properties related to fonts and rendering text |
| **Drawing** | [3.1.1.2.4](#Section_ea08d73894374e80840ad2751ea73a39) | The graphics properties that determine how drawing commands render the image |

The sections that follow describe the elements in each group as well as the EMF records that set their values. Unreferenced intrinsic types are defined in [[MS-DTYP]](%5bMS-DTYP%5d.pdf#Section_cca2742956894a16b2b49325d93e4ba2).

##### Regions

The **Regions** group of elements control the output area and clipping properties in the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45).

| Element name | Type | Set by |
| --- | --- | --- |
| **Clipping** | RegionData (section [2.2.24](#Section_e66601f29b5c46198476ddb7b087551b)) | EMR\_EXCLUDECLIPRECT (section [2.3.2.1](#Section_a521411eb8774199abb6b4514b3574f8))  EMR\_EXTSELECTCLIPRGN (section [2.3.2.2](#Section_c6b9f4e627f64a4da383c2daf5da11d9))  EMR\_INTERSECTCLIPRECT (section [2.3.2.3](#Section_13cd0c98d4e94ca7a79d58055bf45c79))  EMR\_OFFSETCIPRGN (section [2.3.2.4](#Section_8bf2b60d3b194bd1b2d7c89b027ad808)) |
| **MetaClipping** | RegionData | EMR\_SETMETARGN (section [2.3.2](#Section_0ca0d18e324e452f9a4126e1a82e3e03)) |
| **Viewport** | UINT8[16] | EMR\_SCALEVIEWPORTEXTEX (section [2.3.11.7](#Section_469e353c42094919a5aa9331b60765ed))  EMR\_SETVIEWPORTEXTEX (section [2.3.11.28](#Section_a71486a39ebd43e3819f5318a58b70d5))  EMR\_SETVIEWPORTORGEX (section [2.3.11.29](#Section_b89489473f8d4adda4b8e572e8f90506)) |
| **Window** | UINT8[16] | EMR\_SCALEWINDOWEXTEX (section [2.3.11.8](#Section_01698f29bbcb41009bad0fd52c33b0da))  EMR\_SETWINDOWEXTEX (section [2.3.11.30](#Section_1952148360b549b8b58c4b5d1b6cf839))  EMR\_SETWINDOWORGEX (section [2.3.11.31](#Section_952803a93d5c4c83a258558bc1123bdb)) |

**Clipping:** The current clipping [**region**](#gt_6f13fe5f-8747-4ae0-9375-814bf0528197), which with **MetaClipping** defines the bounds of the drawing area. The default value for the **Clipping** element is implementation-specific.[<92>](#Appendix_A_92" \o "Product behavior note 92)

**MetaClipping:** The current metaregion, which with the **Clipping** region defines the bounds of the drawing area.

**Viewport:** A rectangular drawing area using coordinates in the [**device space**](#gt_25d586c7-f440-45c6-b965-5c025bea9fc0).

| Element name | Type | Set by |
| --- | --- | --- |
| **Extent** | SizeL ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.22) | EMR\_SCALEVIEWPORTEXTEX  EMR\_SETVIEWPORTEXTEX |
| **Origin** | PointL ([MS-WMF] section 2.2.2.15) | EMR\_SETVIEWPORTORGEX |

**Extent:** Horizontal and vertical sizes of the drawing area in device units.

**Origin:** Point value of the origin of the drawing area in device units.

**Window:** A rectangular drawing area using the coordinates of the [**page space**](#gt_f530f123-858a-4af7-b626-adea0aeaa75a).

| Element name | Type | Set by |
| --- | --- | --- |
| **Extent** | SizeL | EMR\_SCALEWINDOWEXTEX  EMR\_SETWINDOWEXTEX |
| **Origin** | PointL | EMR\_SETWINDOWORGEX |

**Extent:** Horizontal and vertical sizes of the drawing area in logical units.

**Origin:** Point value of the origin of the drawing area in logical units.

##### Colors

The **Colors** group of elements define the current state of color management in the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45).

| Element name | Type | Set by |
| --- | --- | --- |
| **ColorAdjustment** | ColorAdjustment (section [2.2.2](#Section_71d7f4d0b6454de8a5588a2e0fc596cb)) | EMR\_SETCOLORADJUSTMENT (section [2.3.11.13](#Section_81b3cd7ced6742aaab9847b99d697231)) |
| **ColorProfile** | UINT8[] | EMR\_COLORMATCHTOTARGETW (section [2.3.11.1](#Section_b9e3e8fa1d074475b0f2d157eeca5417))  EMR\_SETICMPROFILEA (section [2.3.11.15](#Section_cf1bc41cbdc54645890fc03e05d4d26d))  EMR\_SETICMPROFILEW (section [2.3.11.16](#Section_1d376b73e0084440b6b95770d4afdee5)) |
| **ColorProfileEmbedded** | BOOL | EMR\_COLORMATCHTOTARGETW |
| **ColorProofing** | UINT32 | EMR\_COLORMATCHTOTARGETW |
| **ColorTransform** | Implementation-dependent | EMR\_COLORMATCHTOTARGETW |
| **ICMMode** | UINT32 | EMR\_SETICMMODE (section [2.3.11.14](#Section_a7c9d4834a2440588dc1b5e0809ea758)) |
| **PixelFormat** | PixelFormatDescriptor (section [2.2.22](#Section_1db036d62da84b92b4f8e9cab8cc93b7)) | EMR\_HEADER (section [2.3.4.2](#Section_de081cd7351f4cc2830bd03fb55e89ab))  EMR\_PIXELFORMAT (section [2.3.11.5](#Section_ba9aa909eb054d2fa51b10cb6df68a54)) |

##### Text

The **Text** group of elements define the current font and text properties in the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45).

**Text** elements are used by the following [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) text drawing records.

* EMR\_EXTTEXTOUTA (section [2.3.5.7](#Section_6b582a713c294fc6a0f41f8a313739a1))
* EMR\_EXTTEXTOUTW (section [2.3.5.8](#Section_a59a79ac328e492da34de02727af6edf))
* EMR\_POLYTEXTOUTA (section [2.3.5.32](#Section_d8b7ac2176b84f9aa7cc05f9d2d5627e))
* EMR\_POLYTEXTOUTW (section [2.3.5.33](#Section_381015eb29a7477493ada210697f5972))
* EMR\_SMALLTEXTOUT (section [2.3.5.37](#Section_20eee81d0bd442d1a624860adfe62358))

| Element name | Type | Set by |
| --- | --- | --- |
| **FontMapperFlags** | UINT32 | EMR\_SETMAPPERFLAGS (section [2.3.11.20](#Section_6647abf7bd7043ef87661a6e06ed2f03)) |
| **ForceUFIMapping** | UniversalFontId (section [2.2.27](#Section_34502dab22b44855b9b288f00a4a0c52)) | EMR\_FORCEUFIMAPPING (section [2.3.11.2](#Section_e9a05924e704403783126bef2e19f6bd)) |
| **LinkedUFIs** | UniversalFontId[] | EMR\_SETLINKEDUFIS (section [2.3.11.18](#Section_3bb62157624f443fa88cef5e696740dd)) |
| **TextAlignment** | UINT32 | EMR\_SETTEXTALIGN (section [2.3.11.25](#Section_a49bb565a58341da8d4a8a0314b3e397)) |
| **TextJustification** | UINT32 [2] | EMR\_SETTEXTJUSTIFICATION (section [2.3.11.27](#Section_edbc2be01da045d69c05677cbcaa2d47)) |

##### Drawing

The **Drawing** group of elements define various graphics flags and other metadata values that affect how the image in the [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7) is rendered.

| Element name | Type | Set by |
| --- | --- | --- |
| **ArcDirection** | UINT32 | EMR\_SETARCDIRECTION (section [2.3.11.9](#Section_83a955e613b84720a521542e3dd91cba)) |
| **BackgroundColor** | ColorRef ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.2.2.8) | EMR\_SETBKCOLOR (section [2.3.11.10](#Section_24f4095262d440b0ba40aad8fb4fc421)) |
| **BackgroundMode** | UINT32 | EMR\_SETBKMODE (section [2.3.11.11](#Section_7c584a42b6dd47b8a0b6f529133292f4)) |
| **BrushOrigin** | PointL | EMR\_SETBRUSHORGEX (section [2.3.11.12](#Section_b02ab7ee4c884bf29ce99100636d86be)) |
| **CurrentPosition** | PointL ([MS-WMF] section 2.2.2.15) | EMR\_MOVETOEX (section [2.3.11.4](#Section_8ae57e5a22e3435298ca09628f47c693)) |
| **LayoutMode** | UINT32 | EMR\_SETLAYOUT (section [2.3.11.17](#Section_dcb69f9d6f69466aaf1e6f505d86cc67)) |
| **LineCap** | UINT32 |  |
| **LineJoin** | UINT32 |  |
| **MappingMode** | UINT32 | EMR\_SETMAPMODE (section [2.3.11.19](#Section_aa4ad35dfa424a4f959a8b41304e1b05)) |
| **MiterLimit** | UINT32 | EMR\_SETMITERLIMIT (section [2.3.11.21](#Section_2637d0d092dd48e8be5e6400b4d1fc5e)) |
| **Path** | PointL[] | Path bracket records (section [2.3.10](#Section_b930b98930ec4954a8891dc60ce0b689)) |
| **PathBracket** | [**Boolean**](#gt_1d79d7a7-ba2c-4b34-931c-7ba8057c87b2) | Path bracket records |
| **PolyFillMode** | UINT32 | EMR\_SETPOLYFILLMODE (section [2.3.11.22](#Section_e8799c80fe7445c1bf6d3eb78edd4ed2)) |
| **ROP2** | UINT32 | EMR\_SETROP2 (section [2.3.11.23](#Section_cedee7e04e6647b6ae1d816e1e319786)) |
| **StretchBLTMode** | UINT32 | EMR\_SETSTRETCHBLTMODE (section [2.3.11.24](#Section_ff00ab986e2c4ecab66404ab2956b6a5)) |
| **TextColor** | ColorRef | EMR\_SETTEXTCOLOR (section [2.3.11.26](#Section_1ae5c616cf8e4f5da7c3bfc1f70ece28)) |

**ArcDirection:** The drawing direction for arcs and rectangles, from the ArcDirection enumeration (section [2.1.2](#Section_9886c62f17884118894a76d5a8733685)).

**BackgroundColor:** The color used as background for drawing text, hatched brushes, and pen styles that are not solid lines, depending on the **BackgroundMode** value.

**BackgroundMode:** How to combine the drawing background with the **BackgroundColor** value, from the BackgroundMode enumeration (section [2.1.4](#Section_2f57054c2b694303a6f85aacd401f759)).

**BrushOrigin:** The horizontal and vertical origin of the current brush in logical units, which is used as needed to maintain an alignment of patterns on the display surface.

### Byte Ordering

The following code snippet illustrates how the use of the [**big-endian**](#gt_6f6f9e8e-5966-4727-8527-7e02fb864e7e) and [**little-endian**](#gt_079478cb-f4c5-4ce5-b72b-2144da5d2ce7) methods can affect the compatibility of applications.

1. #include <unistd.h>
2. #include <sys/stat.h>
3. #include <fcntl.h>
4. int main()
5. {
6. int buf;
7. int in;
8. int nread;
9. in = open("file.in", O\_RDONLY);
10. nread = read(in, (int \*) &buf, sizeof(buf));
11. printf("First Integer in file.in = %x\n", buf);
12. exit(0);
13. }

In the preceding code, if the first integer word stored in the file.in file on a big-endian computer was the hexadecimal number 0x12345678, the resulting output on that computer would be as follows.

1. % ./test
2. First Integer in file.in = 12345678
3. %

If the file.in file were read by the same program running on a little-endian computer, the resulting output would be as follows.

1. % ./test
2. First Integer in file.in = 78563412
3. %

Because of the difference in output, [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7) playback is implemented so that it can read multi-byte values based on the endian method used by the computer that generated the output.

Because metafiles are developed and written on little-endian computers, playback performs this computation on computers that are big-endian.

## EMF Metafile Example

This section describes an example of an [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7), which when processed renders the following image.



Figure 4: EMF metafile example

The contents of this metafile example are shown in this section in hexadecimal bytes. The far-left column is the byte count; the far-right characters are the interpretation of the bytes in the Latin-1 [**ANSI Character Set**](#gt_100cd8a6-5cb1-4895-9de6-e4a3c224a583) [[ISO/IEC-8859-1]](https://go.microsoft.com/fwlink/?LinkId=90689). The sections that follow describe the packets that convey this series of bytes.

1. 00000000:01 00 00 00 D4 00 00 00 00 00 00 00 00 00 00 00 ....Ô...........
2. 00000010:59 00 00 00 59 00 00 00 00 00 00 00 00 00 00 00 Y...Y...........
3. 00000020:42 0C 00 00 41 0C 00 00 20 45 4D 46 00 00 01 00 B...A... EMF....
4. 00000030:FC 37 00 00 16 00 00 00 05 00 00 00 34 00 00 00 ü7..........4...
5. 00000040:6C 00 00 00 00 00 00 00 80 07 00 00 B0 04 00 00 l.......€...°...
6. 00000050:A5 02 00 00 A7 01 00 00 00 00 00 00 00 00 00 00 ¥...§...........
7. 00000060:00 00 00 00 D5 55 0A 00 A5 75 06 00 53 00 61 00 ....ÕU..¥u..S.a.
8. 00000070:6D 00 70 00 6C 00 65 00 20 00 45 00 4D 00 46 00 m.p.l.e. .E.M.F.
9. 00000080:20 00 74 00 68 00 61 00 74 00 20 00 68 00 61 00 .t.h.a.t. .h.a.
10. 00000090:73 00 20 00 61 00 20 00 62 00 72 00 75 00 73 00 s. .a. .b.r.u.s.
11. 000000A0:68 00 20 00 66 00 69 00 6C 00 6C 00 2C 00 20 00 h. .f.i.l.l.,. .
12. 000000B0:62 00 69 00 74 00 6D 00 61 00 70 00 2C 00 20 00 b.i.t.m.a.p.,. .
13. 000000C0:61 00 6E 00 64 00 20 00 74 00 65 00 78 00 74 00 a.n.d. .t.e.x.t.
14. 000000D0:00 00 00 00 27 00 00 00 18 00 00 00 01 00 00 00 ....'...........
15. 000000E0:02 00 00 00 52 47 2A 00 03 00 00 00 25 00 00 00 ....RG\*.....%...
16. 000000F0:0C 00 00 00 01 00 00 00 4C 00 00 00 64 00 00 00 ........L...D...
17. 00000100:00 00 00 00 00 00 00 00 59 00 00 00 59 00 00 00 ........Y...Y...
18. 00000110:00 00 00 00 00 00 00 00 5A 00 00 00 5A 00 00 00 ........Z...Z...
19. 00000120:21 00 F0 00 00 00 00 00 00 00 00 00 00 00 80 3F !.ð...........€?
20. 00000130:00 00 00 00 00 00 00 00 00 00 80 3F 00 00 00 00 ..........€?....
21. 00000140:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
22. 00000150:00 00 00 00 00 00 00 00 00 00 00 00 25 00 00 00 ............%...
23. 00000160:0C 00 00 00 00 00 00 80 4C 00 00 00 A8 2F 00 00 .......€L...¨/..
24. 00000170:00 00 00 00 2D 00 00 00 59 00 00 00 59 00 00 00 ....-...Y...Y...
25. 00000180:00 00 00 00 2D 00 00 00 5A 00 00 00 2D 00 00 00 ....-...Z...-...
26. 00000190:20 00 CC 00 00 00 00 00 00 00 00 00 00 00 80 3F .Ì...........€?
27. 000001A0:00 00 00 00 00 00 00 00 00 00 80 3F 00 00 00 00 ..........€?....
28. 000001B0:00 00 00 00 FF FF FF 00 00 00 00 00 64 00 00 00 ....ÿÿÿ.....D...
29. 000001C0:28 00 00 00 8C 00 00 00 1C 2F 00 00 28 00 00 00 (...Œ..../..(...
30. 000001D0:59 00 00 00 2D 00 00 00 01 00 18 00 00 00 00 00 Y...-...........
31. 000001E0:1C 2F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ./..............
32. 000001F0:00 00 00 00 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E ....?\_.?\_.?\_.?\_.
33. 00000200:3F 5F 0E 3F 5F 0E 3F 5F 0E 3E 5E 0D 3F 5F 0E 3F ?\_.?\_.?\_.>^.?\_.?
34. 00000210:5F 0E 41 5E 0E 41 5E 0E 41 5E 0E 41 5E 0E 41 5E \_.A^.A^.A^.A^.A^
35. 00000220:0E 41 5E 0E 41 5E 0E 41 5E 0E 41 5E 0E 41 5E 0E .A^.A^.A^.A^.A^.
36. 00000230:41 5E 0E 41 5E 0E 41 5E 0E 41 5E 0E 40 5D 0D 3D A^.A^.A^.A^.@].=
37. 00000240:59 0C 5A 60 4E AE AE AE BF BF BF C5 C5 C5 C0 C0 Y.Z`N®®®¿¿¿ÅÅÅÀÀ
38. 00000250:C1 B1 B2 B1 A1 A1 A1 A9 AA AA AE AF B0 A2 A2 A2 Á±²±¡¡¡©ªª®¯°¢¢¢
39. 00000260:A6 A6 A4 AF AE AD AC AC AC A6 A6 A6 99 99 99 7D ¦¦¤¯®­¬¬¬¦¦¦™™™}
40. 00000270:7D 7D 66 66 65 5A 5A 59 4F 4F 4F 58 58 58 76 76 }}FFEZZYOOOXXXvv
41. 00000280:76 9E 9E 9E B5 B5 B5 3C 43 2D 32 46 0A 34 4A 0A vžžžµµµ<C-2F.4J.
42. 00000290:34 49 0A 36 4C 0B 3A 52 0B 3F 59 0C 41 5C 0D 42 4I.6L.:R.?Y.A\.B
43. 000002A0:5D 0D 42 5D 0D 44 5D 0D 44 5D 0D 43 5C 0C 43 5C ].B].D].D].C\.C\
44. 000002B0:0C 43 5C 0C 42 5B 0B 43 5C 0C 43 5C 0C 43 5C 0C .C\.B[.C\.C\.C\.
45. 000002C0:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 C\.C\.C\.C\.C\.C
46. 000002D0:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C \.C\.C\.C\.C\.C\
47. 000002E0:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C .C\.C\.C\.C\.C\.
48. 000002F0:45 5B 0C 45 5B 0C 45 5C 0A 45 5C 0A 45 5C 0A 00 E[.E[.E\.E\.E\..
49. 00000300:3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F ?\_.?\_.?\_.?\_.?\_.?
50. 00000310:5F 0E 3F 5F 0E 3E 5E 0D 3F 5F 0E 3F 5F 0E 41 5E \_.?\_.>^.?\_.?\_.A^
51. 00000320:0E 41 5E 0E 41 5E 0E 41 5E 0E 41 5E 0E 41 5E 0E .A^.A^.A^.A^.A^.
52. 00000330:41 5E 0E 41 5E 0E 41 5E 0E 41 5E 0E 41 5E 0E 41 A^.A^.A^.A^.A^.A
53. 00000340:5E 0E 41 5E 0E 41 5E 0E 40 5D 0D 39 52 0C 70 73 ^.A^.A^.@].9R.ps.
54. 00000350:6B AD AD AD A8 A8 A8 99 99 98 A6 A6 A6 A6 A7 A7 k­­­¨¨¨™™˜¦¦¦¦§§
55. 00000360:A3 A5 A6 A6 A5 A3 AE A5 9D C9 BB AD A1 9B 95 A2 £¥¦¦¥£®¥□É»­¡›•¢
56. 00000370:9F 9C B3 B1 AF B6 B4 B3 A9 A9 A9 97 93 90 8E 87 Ÿœ³±¯¶´³©©©--"□Ž‡
57. 00000380:81 89 84 7F 81 7E 7A 79 78 76 80 7F 7E 98 98 98 □‰„□~zyxv€~˜˜˜
58. 00000390:BC BC BC 6D 70 65 33 47 0A 37 4D 0B 35 4B 0A 35 ¼¼¼mpE3G.7M.5K.5
59. 000003A0:4B 0B 38 4F 0B 3D 56 0C 41 5B 0D 42 5D 0D 42 5D K.8O.=V.A[.B].B]
60. 000003B0:0D 43 5D 0D 44 5D 0D 43 5C 0C 43 5C 0C 43 5C 0C .C].D].C\.C\.C\.
61. 000003C0:42 5B 0B 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 B[.C\.C\.C\.C\.C
62. 000003D0:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C \.C\.C\.C\.C\.C\
63. 000003E0:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C .C\.C\.C\.C\.C\.
64. 000003F0:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 45 5B 0C 45 C\.C\.C\.C\.E[.E
65. 00000400:5B 0C 45 5C 0A 45 5C 0A 45 5C 0A 00 3F 5F 0E 3F [.E\.E\.E\..?\_.?
66. 00000410:5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F \_.?\_.?\_.?\_.?\_.?\_
67. 00000420:0E 3E 5E 0D 3F 5F 0E 3F 5F 0E 41 5E 0E 41 5E 0E .>^.?\_.?\_.A^.A^.
68. 00000430:41 5E 0E 41 5E 0E 41 5E 0E 41 5E 0E 41 5E 0E 41 A^.A^.A^.A^.A^.A
69. 00000440:5E 0E 41 5E 0E 41 5E 0E 41 5E 0E 41 5E 0E 41 5E ^.A^.A^.A^.A^.A^
70. 00000450:0E 41 5E 0E 40 5D 0D 38 51 0E 90 91 8E AC AC AC .A^.@].8Q.□'Ž¬¬¬
71. 00000460:C3 C3 C3 C5 C6 C6 B9 BB BC B3 B0 AD BA AD 9E C8 ÃÃÃÅÆÆ¹»¼³°­º­žÈ
72. 00000470:B4 A0 DB C8 B4 E4 D1 BE C2 B5 A9 82 7E 7A 8C 88 ´ ÛÈ´äÑ¾Âµ©‚~zŒˆ
73. 00000480:84 83 80 7E 98 8F 85 BE AC 9A C9 B6 A3 CC BB A9 „ƒ€~˜□…¾¬šÉ¶£Ì»©
74. 00000490:CD C0 B3 AB A3 9B 80 7E 76 78 76 70 91 8F 8D 84 ÍÀ³«£›€~vxvp'□□„
75. 000004A0:86 80 36 4B 0B 39 50 0B 38 4E 0B 37 4D 0B 38 4F †€6K.9P.8N.7M.8O
76. 000004B0:0B 3C 55 0C 40 5A 0D 42 5D 0D 42 5D 0D 42 5D 0D .<U.@Z.B].B].B].
77. 000004C0:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 42 5B 0B 43 C\.C\.C\.C\.B[.C
78. 000004D0:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C \.C\.C\.C\.C\.C\
79. 000004E0:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C .C\.C\.C\.C\.C\.
80. 000004F0:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 C\.C\.C\.C\.C\.C
81. 00000500:5C 0C 43 5C 0C 43 5C 0C 45 5B 0A 45 5B 0A 45 5C \.C\.C\.E[.E[.E\
82. 00000510:0A 45 5C 0A 45 5C 0A 00 3F 5F 0E 3F 5F 0E 3F 5F .E\.E\..?\_.?\_.?\_
83. 00000520:0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3E 5E 0D .?\_.?\_.?\_.?\_.>^.
84. 00000530:3F 5F 0E 3F 5F 0E 41 5E 0E 41 5E 0E 41 5E 0E 41 ?\_.?\_.A^.A^.A^.A
85. 00000540:5E 0E 41 5E 0E 41 5E 0E 41 5E 0E 41 5E 0E 41 5E ^.A^.A^.A^.A^.A^
86. 00000550:0E 41 5E 0E 41 5E 0E 41 5E 0E 41 5E 0E 41 5E 0E .A^.A^.A^.A^.A^.
87. 00000560:40 5D 0D 3D 50 1C 94 94 94 A1 A1 A0 C9 CA CA BC @].=P."""¡¡ ÉÊÊ¼
88. 00000570:BB B8 BB AC 9C C4 AD 94 D2 BB A4 E0 CF BD E0 D0 »¸»¬œÄ­"Ò»¤àÏ½àÐ
89. 00000580:BF DA C7 B5 DE CB B8 9A 93 8B 7C 76 70 A2 93 83 ¿ÚÇµÞË¸š"‹|vp¢"ƒ
90. 00000590:C0 AB 95 C6 B2 9D CA B8 A4 D1 C0 AD E0 D2 C4 E8 À«•Æ²□Ê¸¤ÑÀ­àÒÄè
91. 000005A0:DC CF CD C3 B5 8A 85 74 72 72 61 83 82 7E 34 43 ÜÏÍÃµŠ…trrAƒ‚~4C
92. 000005B0:15 3C 54 0C 38 50 0B 38 4E 0B 39 50 0B 3D 55 0C .<T.8P.8N.9P.=U.
93. 000005C0:40 5B 0D 42 5D 0D 42 5D 0D 42 5D 0D 43 5C 0C 43 @[.B].B].B].C\.C
94. 000005D0:5C 0C 43 5C 0C 43 5C 0C 42 5B 0B 43 5C 0C 43 5C \.C\.C\.B[.C\.C\
95. 000005E0:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C .C\.C\.C\.C\.C\.
96. 000005F0:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 C\.C\.C\.C\.C\.C
97. 00000600:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C \.C\.C\.C\.C\.C\
98. 00000610:0C 43 5C 0A 45 5C 0A 45 5C 0A 45 5C 0A 45 5C 0A .C\.E\.E\.E\.E\.
99. 00000620:45 5C 0A 00 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E E\..?\_.?\_.?\_.?\_.
100. 00000630:3F 5F 0E 3F 5F 0E 3F 5F 0E 3E 5E 0D 3F 5F 0E 3F ?\_.?\_.?\_.>^.?\_.?
101. 00000640:5F 0E 41 5E 0E 41 5E 0E 41 5E 0E 41 5E 0E 41 5E \_.A^.A^.A^.A^.A^
102. 00000650:0E 41 5E 0E 41 5E 0E 41 5E 0E 41 5E 0E 41 5E 0E .A^.A^.A^.A^.A^.
103. 00000660:41 5E 0E 41 5E 0E 41 5E 0E 41 5E 0E 40 5D 0D 4B A^.A^.A^.A^.@].K
104. 00000670:56 33 B8 B8 B8 B7 B7 B7 A1 A1 A2 BC AC 9B CC B2 V3¸¸¸•••¡¡¢¼¬›Ì²
105. 00000680:97 D6 C3 AE E5 D6 C6 E4 D4 C4 E1 D2 C2 D9 C6 B4 --ÖÃ®åÖÆäÔÄáÒÂÙÆ´
106. 00000690:D7 C3 AD CE BF AF AD 9A 85 BB A5 8D BD A9 93 C3 ×Ã­Î¿¯­š…»¥□½©"Ã
107. 000006A0:AF 9B C8 B5 A2 CE BD AA DC CE C0 E2 D5 C9 E1 D4 ¯›Èµ¢Î½ªÜÎÀâÕÉáÔ
108. 000006B0:C7 D5 C5 B4 B1 A9 95 91 97 7B 4A 57 2D 39 51 0B ÇÕÅ´±©•'--{JW-9Q.
109. 000006C0:38 50 0B 38 50 0B 3A 52 0C 3E 58 0C 41 5C 0D 42 8P.8P.:R.>X.A\.B
110. 000006D0:5D 0D 42 5D 0D 42 5D 0D 41 5C 0C 43 5C 0C 43 5C ].B].B].A\.C\.C\
111. 000006E0:0C 43 5C 0C 42 5B 0B 43 5C 0C 43 5C 0C 43 5C 0C .C\.B[.C\.C\.C\.
112. 000006F0:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 C\.C\.C\.C\.C\.C
113. 00000700:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C \.C\.C\.C\.C\.C\
114. 00000710:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5D 0A .C\.C\.C\.C\.C].
115. 00000720:45 5C 0A 45 5C 0A 45 5C 0A 45 5C 0A 45 5C 0A 00 E\.E\.E\.E\.E\..
116. 00000730:3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F ?\_.?\_.?\_.?\_.?\_.?
117. 00000740:5F 0E 3F 5F 0E 40 5D 0D 3F 5F 0E 3F 5F 0E 41 5E \_.?\_.@].?\_.?\_.A^
118. 00000750:0F 41 5E 0F 41 5E 0F 41 5E 0E 41 5E 0E 40 5D 0D .A^.A^.A^.A^.@].
119. 00000760:40 5D 0D 40 5D 0D 41 5E 0E 40 5D 0D 40 5D 0D 40 @].@].A^.@].@].@
120. 00000770:5D 0D 40 5D 0D 41 5E 0E 41 5E 0E 61 6C 44 D0 D0 ].@].A^.A^.AlDÐÐ
121. 00000780:D0 B0 B0 B0 C8 C3 BE DB C8 B3 DC CA B8 EB DD CF Ð°°°ÈÃ¾ÛÈ³ÜÊ¸ëÝÏ
122. 00000790:E8 D9 CA E5 D6 C7 E4 D6 C8 D9 C8 B5 D2 BF AA DD èÙÊåÖÇäÖÈÙÈµÒ¿ªÝ
123. 000007A0:CC BA C7 B3 9E B6 9F 89 BC A7 92 C1 AD 98 C6 B3 ÌºÇ³ž¶Ÿ‰¼§'Á­˜Æ³
124. 000007B0:9F CC BA A8 DB CD BF DD D0 C2 DF D3 C6 E0 D3 C5 ŸÌº¨ÛÍ¿ÝÐÂßÓÆàÓÅ
125. 000007C0:E2 D4 C4 A7 AA 8E 52 62 2B 37 4D 0B 39 51 0B 3B âÔÄ§ªŽRB+7M.9Q.;
126. 000007D0:53 0C 3E 57 0C 40 5A 0D 42 5D 0D 42 5D 0D 42 5D S.>W.@Z.B].B].B]
127. 000007E0:0D 42 5D 0D 44 5D 0D 43 5C 0C 43 5C 0C 45 5B 0D .B].D].C\.C\.E[.
128. 000007F0:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 C\.C\.C\.C\.C\.C
129. 00000800:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C \.C\.C\.C\.C\.C\
130. 00000810:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C .C\.C\.C\.C\.C\.
131. 00000820:43 5C 0C 43 5C 0C 43 5C 0C 45 5C 0A 45 5C 0A 45 C\.C\.C\.E\.E\.E
132. 00000830:5C 0A 45 5C 0A 45 5C 0A 45 5B 0C 00 3F 5F 0E 3F \.E\.E\.E[..?\_.?
133. 00000840:5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F \_.?\_.?\_.?\_.?\_.?\_
134. 00000850:0E 40 5D 0D 3F 5F 0E 3F 5F 0E 41 5E 0E 41 5E 0F .@].?\_.?\_.A^.A^.
135. 00000860:41 5E 0E 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 A^.@].@].@].@].@
136. 00000870:5D 0D 41 5E 0E 40 5D 0D 40 5D 0D 40 5D 0D 40 5D ].A^.@].@].@].@]
137. 00000880:0D 41 5E 0E 3F 5B 0D 6A 79 47 D1 D2 CE C5 C2 BF .A^.?[.jyGÑÒÎÅÂ¿
138. 00000890:EF E6 DD E7 D8 CA EE E0 D3 ED DF D2 EA DC CE E9 ïæÝçØÊîàÓíßÒêÜÎé
139. 000008A0:DB CD E1 D1 C1 D6 C3 B0 D7 C4 B1 DA C9 B8 D5 C3 ÛÍáÑÁÖÃ°×Ä±ÚÉ¸ÕÃ
140. 000008B0:B1 B8 A3 8C BC A7 91 C0 AC 97 C5 B1 9D CB B9 A5 ±¸£Œ¼§'À¬--Å±□Ë¹¥
141. 000008C0:D6 C8 B8 CE BF AE D9 CB BD E1 D4 C7 E3 D4 C6 BC ÖÈ¸Î¿®ÙË½áÔÇãÔÆ¼
142. 000008D0:B9 A2 53 63 2C 37 4C 0B 3C 55 0C 3E 58 0C 40 5B ¹¢SC,7L.<U.>X.@[
143. 000008E0:0D 41 5C 0D 42 5D 0D 42 5D 0D 42 5D 0D 42 5D 0D .A\.B].B].B].B].
144. 000008F0:43 5C 0C 43 5C 0C 43 5C 0C 45 5B 0C 43 5C 0C 43 C\.C\.C\.E[.C\.C
145. 00000900:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C \.C\.C\.C\.C\.C\
146. 00000910:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C .C\.C\.C\.C\.C\.
147. 00000920:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 C\.C\.C\.C\.C\.C
148. 00000930:5C 0C 43 5C 0C 45 5C 0A 45 5C 0A 45 5C 0A 45 5C \.C\.E\.E\.E\.E\
149. 00000940:0A 45 5C 0A 45 5B 0C 00 3F 5F 0E 3F 5F 0E 3F 5F .E\.E[..?\_.?\_.?\_
150. 00000950:0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 40 5D 0D .?\_.?\_.?\_.?\_.@].
151. 00000960:3F 5F 0E 3F 5F 0E 41 5E 0E 41 5E 0E 40 5D 0D 40 ?\_.?\_.A^.A^.@].@
152. 00000970:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 41 5E ].@].@].@].@].A^
153. 00000980:0E 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 41 5E 0E .@].@].@].@].A^.
154. 00000990:40 59 0F 75 84 54 A6 B1 8E DC D9 CE F5 EE E8 E7 @Y.u„T¦±ŽÜÙÎõîèç
155. 000009A0:DA CE E5 D7 C8 EB DD D0 EC DF D2 E4 D5 C4 D8 C8 ÚÎå×ÈëÝÐìßÒäÕÄØÈ
156. 000009B0:B5 D9 CA B8 D9 CB BA D9 C8 B6 D9 C8 B6 C4 B0 9C µÙÊ¸ÙËºÙÈ¶ÙÈ¶Ä°œ
157. 000009C0:BA A5 90 BF AA 95 C3 AF 9B C9 B6 A3 C9 B9 A7 C7 º¥□¿ª•Ã¯›É¶£É¹§Ç
158. 000009D0:B7 A6 D1 C2 B3 DC CF C0 E0 D2 C3 CE C4 B3 5B 69 •¦ÑÂ³ÜÏÀàÒÃÎÄ³[i
159. 000009E0:33 3C 52 0D 40 5A 0D 41 5C 0D 42 5D 0D 42 5D 0D 3<R.@Z.A\.B].B].
160. 000009F0:42 5D 0D 42 5D 0D 41 5C 0C 42 5D 0D 43 5C 0C 43 B].B].A\.B].C\.C
161. 00000A00:5C 0C 43 5C 0C 45 5B 0C 43 5C 0C 43 5C 0C 43 5C \.C\.E[.C\.C\.C\
162. 00000A10:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C .C\.C\.C\.C\.C\.
163. 00000A20:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 C\.C\.C\.C\.C\.C
164. 00000A30:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C \.C\.C\.C\.C\.C\
165. 00000A40:0C 45 5C 0A 45 5C 0A 45 5C 0A 45 5C 0A 45 5C 0A .E\.E\.E\.E\.E\.
166. 00000A50:45 5B 0C 00 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E E[..?\_.?\_.?\_.?\_.
167. 00000A60:3F 5F 0E 3F 5F 0E 3F 5F 0E 40 5D 0D 3F 5F 0E 3F ?\_.?\_.?\_.@].?\_.?
168. 00000A70:5F 0E 41 5E 0E 41 5E 0E 40 5D 0D 40 5D 0D 40 5D \_.A^.A^.@].@].@]
169. 00000A80:0D 40 5D 0D 40 5D 0D 40 5D 0D 41 5E 0E 40 5D 0D .@].@].@].A^.@].
170. 00000A90:40 5D 0D 40 5D 0D 40 5D 0D 41 5E 0E 44 5C 12 8B @].@].@].A^.D\.‹
171. 00000AA0:97 6A 93 A2 76 C8 CA B5 F5 EE E9 E8 DC CF E0 D0 --j"¢vÈÊµõîéèÜÏàÐ
172. 00000AB0:C1 DF D2 C4 E1 BB A1 D4 AC 8E D5 B2 97 D3 A4 84 ÁßÒÄá»¡Ô¬ŽÕ²--Ó¤„
173. 00000AC0:D3 A5 86 D7 C4 B0 D7 C5 B1 CF BB A6 C2 AE 99 C1 Ó¥†×Ä°×Å±Ï»¦Â®™Á
174. 00000AD0:AD 98 C1 AE 99 C5 B0 9B C7 B5 A2 C8 B8 A7 D1 C2 ­˜Á®™Å°›Çµ¢È¸§ÑÂ
175. 00000AE0:B2 E0 D3 C6 E4 D8 CA E2 D5 C6 71 7B 4B 42 58 10 ²àÓÆäØÊâÕÆq{KBX.
176. 00000AF0:42 5D 0D 42 5D 0D 42 5D 0D 42 5D 0D 42 5D 0D 42 B].B].B].B].B].B
177. 00000B00:5D 0D 41 5C 0C 41 5C 0C 43 5C 0C 43 5C 0C 43 5C ].A\.A\.C\.C\.C\
178. 00000B10:0C 45 5B 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C .E[.C\.C\.C\.C\.
179. 00000B20:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 C\.C\.C\.C\.C\.C
180. 00000B30:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C \.C\.C\.C\.C\.C\
181. 00000B40:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 45 5C 0A .C\.C\.C\.C\.E\.
182. 00000B50:45 5C 0A 45 5C 0A 45 5C 0A 45 5C 0A 45 5B 0C 00 E\.E\.E\.E\.E[..
183. 00000B60:3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F ?\_.?\_.?\_.?\_.?\_.?
184. 00000B70:5F 0E 3F 5F 0E 40 5D 0D 3F 5F 0E 3F 5F 0E 41 5E \_.?\_.@].?\_.?\_.A^
185. 00000B80:0E 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D .@].@].@].@].@].
186. 00000B90:40 5D 0D 40 5D 0D 41 5E 0E 40 5D 0D 40 5D 0D 40 @].@].A^.@].@].@
187. 00000BA0:5D 0D 40 5D 0D 41 5E 0E 4B 60 17 A0 AB 83 92 A2 ].@].A^.K`. «ƒ'¢
188. 00000BB0:73 AC B6 94 F4 ED E7 E2 CF BE DA CA BA D7 C4 B2 s¬¶"ôíçâÏ¾ÚÊº×Ä²
189. 00000BC0:D0 86 58 CA 71 3D C9 69 32 C6 5D 24 C4 54 17 CC Ð†XÊq=Éi2Æ]$ÄT.Ì
190. 00000BD0:86 5D D5 C5 B2 D2 BC A6 C9 B3 9D C2 AD 96 C1 AC †]ÕÅ²Ò¼¦É³□Â­–Á¬
191. 00000BE0:97 C8 B1 9A D9 C8 B6 D3 C4 B4 D1 C3 B3 DE D1 C4 --È±šÙÈ¶ÓÄ´ÑÃ³ÞÑÄ
192. 00000BF0:E0 D3 C5 E3 D5 C6 A3 A4 81 48 5E 13 42 5D 0D 42 àÓÅãÕÆ£¤□H^.B].B
193. 00000C00:5D 0D 42 5D 0D 42 5D 0D 42 5D 0D 42 5D 0D 41 5C ].B].B].B].B].A\
194. 00000C10:0C 41 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 45 5B 0C .A\.C\.C\.C\.E[.
195. 00000C20:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 C\.C\.C\.C\.C\.C
196. 00000C30:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C \.C\.C\.C\.C\.C\
197. 00000C40:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C .C\.C\.C\.C\.C\.
198. 00000C50:43 5C 0C 43 5C 0C 43 5C 0C 45 5C 0A 45 5C 0A 45 C\.C\.C\.E\.E\.E
199. 00000C60:5C 0A 45 5C 0A 45 5C 0A 45 5B 0C 00 3F 5F 0E 3F \.E\.E\.E[..?\_.?
200. 00000C70:5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F \_.?\_.?\_.?\_.?\_.?\_
201. 00000C80:0E 40 5D 0D 3E 5E 0D 3E 5E 0D 40 5D 0D 40 5D 0D .@].>^.>^.@].@].
202. 00000C90:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 @].@].@].@].@].@
203. 00000CA0:5D 0D 41 5E 0E 40 5D 0D 40 5D 0D 40 5D 0D 40 5D ].A^.@].@].@].@]
204. 00000CB0:0D 41 5E 0E 52 64 1F A4 AF 8A 95 A3 75 B3 AC 84 .A^.RD.¤¯Š•£u³¬„
205. 00000CC0:DC AB 87 D7 A3 7D DB CD BD D2 A1 7F CD 76 42 CB Ü«‡×£}ÛÍ½Ò¡ÍvBË
206. 00000CD0:71 3C C9 69 32 C7 62 2A C5 5B 20 C3 54 19 D8 AD q<Éi2ÇB\*Å[ ÃT.Ø­
207. 00000CE0:90 E2 D7 CA D3 BE A9 CA B6 A0 D0 BD AA D1 BD A9 □â×ÊÓ¾©Ê¶ Ð½ªÑ½©
208. 00000CF0:DD CD BD DD D0 C2 D0 C1 B1 DA CD BE DE D1 C3 DB ÝÍ½ÝÐÂÐÁ±ÚÍ¾ÞÑÃÛ
209. 00000D00:CD BC 91 9A 6E 4E 62 18 42 5D 0D 42 5D 0D 42 5D Í¼'šnNB.B].B].B]
210. 00000D10:0D 42 5D 0D 42 5E 0B 41 5C 0C 41 5C 0C 41 5C 0C .B].B^.A\.A\.A\.
211. 00000D20:43 5C 0C 43 5C 0C 43 5C 0C 45 5B 0C 43 5C 0C 43 C\.C\.C\.E[.C\.C
212. 00000D30:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C \.C\.C\.C\.C\.C\
213. 00000D40:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C .C\.C\.C\.C\.C\.
214. 00000D50:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 C\.C\.C\.C\.C\.C
215. 00000D60:5C 0C 43 5C 0C 45 5C 0A 45 5C 0A 45 5C 0A 45 5C \.C\.E\.E\.E\.E\
216. 00000D70:0A 45 5C 0A 45 5B 0C 00 3F 5F 0E 3F 5F 0E 3F 5F .E\.E[..?\_.?\_.?\_
217. 00000D80:0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 40 5D 0D .?\_.?\_.?\_.?\_.@].
218. 00000D90:3E 5D 0D 3E 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 >].>].@].@].@].@
219. 00000DA0:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D ].@].@].@].@].@]
220. 00000DB0:0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0E .@].@].@].@].@].
221. 00000DC0:58 67 26 A4 B1 8B B8 B1 8A D6 9F 74 D5 98 6B D7 Xg&¤±‹¸±ŠÖŸtÕ˜k×
222. 00000DD0:A4 7F DD CF BF D2 90 66 CD 77 43 CA 6E 38 C8 65 ¤ÝÏ¿Ò□FÍwCÊn8ÈE
223. 00000DE0:2C C6 60 27 C6 5B 21 C3 56 1B CB 71 40 DD C8 B6 ,Æ`'Æ[!ÃV.Ëq@ÝÈ¶
224. 00000DF0:D5 C3 AF E1 D2 C2 DF D0 C1 D9 C8 B7 E1 D3 C4 E4 ÕÃ¯áÒÂßÐÁÙÈ•áÓÄä
225. 00000E00:D8 CA D1 C2 B2 E0 D4 C7 E7 DA CD AD AA 90 6F 7F ØÊÑÂ²àÔÇçÚÍ­ª□o
226. 00000E10:47 52 63 1F 41 5B 0D 41 5C 0C 41 5C 0C 42 5D 0D GRC.A[.A\.A\.B].
227. 00000E20:42 5D 0B 41 5C 0C 41 5C 0C 41 5B 0C 42 5B 0C 43 B].A\.A\.A[.B[.C
228. 00000E30:5B 0C 43 5C 0C 43 5B 0B 43 5C 0C 43 5C 0C 43 5B [.C\.C[.C\.C\.C[
229. 00000E40:0C 43 5B 0C 43 5B 0C 43 5C 0C 43 5C 0C 43 5C 0C .C[.C[.C\.C\.C\.
230. 00000E50:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 C\.C\.C\.C\.C\.C
231. 00000E60:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C \.C\.C\.C\.C\.C\
232. 00000E70:0C 44 5C 0A 44 5C 0A 45 5B 0A 45 5B 0A 45 5B 0A .D\.D\.E[.E[.E[.
233. 00000E80:45 5B 0B 00 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E E[..?\_.?\_.?\_.?\_.
234. 00000E90:3F 5F 0E 3F 5F 0E 3F 5F 0E 40 5D 0D 40 5D 0D 40 ?\_.?\_.?\_.@].@].@
235. 00000EA0:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D ].@].@].@].@].@]
236. 00000EB0:0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 41 5E 0E .@].@].@].@].A^.
237. 00000EC0:40 5D 0D 40 5D 0D 40 5D 0D 42 5C 0E 60 6E 33 A9 @].@].@].B\.`n3©
238. 00000ED0:B6 91 CD B0 8B D7 A2 78 D6 99 6D D9 A8 86 E2 D7 ¶'Í°‹×¢xÖ™mÙ¨†â×
239. 00000EE0:CA DE BC A2 CC 78 44 D0 85 57 D2 8B 61 CC 74 42 ÊÞ¼¢ÌxDÐ…WÒ‹AÌtB
240. 00000EF0:CC 76 46 CD 7D 4F CC 76 47 CF 9C 7A D8 C9 B7 DF ÌvFÍ}OÌvGÏœzØÉ•ß
241. 00000F00:D0 C0 DF CF BF DD CE BE E4 D7 C9 EB E0 D4 D1 C3 ÐÀßÏ¿ÝÎ¾ä×ÉëàÔÑÃ
242. 00000F10:B3 DC CF C1 F1 E6 DA C8 C0 AE 72 80 4F 56 66 24 ³ÜÏÁñæÚÈÀ®r€OVF$
243. 00000F20:42 5B 0E 41 5C 0C 41 5C 0C 42 5D 0D 43 5C 0C 43 B[.A\.A\.B].C\.C
244. 00000F30:5C 0C 43 5C 0C 43 5B 0D 42 5A 0C 43 5B 0D 43 5C \.C\.C[.BZ.C[.C\
245. 00000F40:0C 42 5B 0B 43 5C 0C 43 5C 0C 43 5B 0D 43 5B 0D .B[.C\.C\.C[.C[.
246. 00000F50:43 5B 0D 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 C[.C\.C\.C\.C\.C
247. 00000F60:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C \.C\.C\.C\.C\.C\
248. 00000F70:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C .C\.C\.C\.C\.C\.
249. 00000F80:43 5C 0C 45 5B 0C 45 5B 0C 45 5B 0C 45 5C 0A 00 C\.E[.E[.E[.E\..
250. 00000F90:3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F ?\_.?\_.?\_.?\_.?\_.?
251. 00000FA0:5F 0E 3F 5F 0E 40 5D 0D 40 5D 0D 40 5D 0D 40 5D \_.?\_.@].@].@].@]
252. 00000FB0:0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D .@].@].@].@].@].
253. 00000FC0:40 5D 0D 40 5D 0D 40 5D 0D 41 5E 0E 40 5D 0D 40 @].@].@].A^.@].@
254. 00000FD0:5D 0D 40 5D 0D 43 5C 10 78 82 4C B9 BF 9C D8 AB ].@].C\.x‚L¹¿œØ«
255. 00000FE0:84 D6 A3 7A D6 A2 7C DE BE A5 E7 DA CD E8 DA CB „Ö£zÖ¢|Þ¾¥çÚÍèÚË
256. 00000FF0:DA AA 89 E7 D9 CC DB AE 91 C7 63 2B C6 5D 24 C4 Úª‰çÙÌÛ®'ÇC+Æ]$Ä
257. 00001000:56 1B C2 51 15 D1 A2 83 E1 D4 C5 E1 D2 C2 E1 D1 V.ÂQ.Ñ¢ƒáÔÅáÒÂáÑ
258. 00001010:C2 D1 C0 AD D1 C2 B1 E2 D6 C9 D3 C5 B5 D4 C6 B7 ÂÑÀ­ÑÂ±âÖÉÓÅµÔÆ•
259. 00001020:E2 D6 C8 EA DE D2 95 9A 76 5B 6B 2A 44 5C 0F 41 âÖÈêÞÒ•šv[k\*D\.A
260. 00001030:5C 0C 41 5C 0C 41 5C 0C 43 5C 0C 43 5C 0C 43 5C \.A\.A\.C\.C\.C\
261. 00001040:0C 43 5B 0D 42 5A 0C 43 5B 0D 43 5C 0C 42 5B 0B .C[.BZ.C[.C\.B[.
262. 00001050:43 5C 0C 43 5C 0C 43 5B 0D 43 5B 0D 43 5B 0D 43 C\.C\.C[.C[.C[.C
263. 00001060:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C \.C\.C\.C\.C\.C\
264. 00001070:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C .C\.C\.C\.C\.C\.
265. 00001080:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 45 C\.C\.C\.C\.C\.E
266. 00001090:5B 0C 45 5B 0C 45 5B 0C 45 5C 0A 00 3F 5F 0E 3F [.E[.E[.E\..?\_.?
267. 000010A0:5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F \_.?\_.?\_.?\_.?\_.?\_
268. 000010B0:0E 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D .@].@].@].@].@].
269. 000010C0:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 @].@].@].@].@].@
270. 000010D0:5D 0D 40 5D 0D 41 5E 0E 40 5D 0D 40 5D 0D 40 5D ].@].A^.@].@].@]
271. 000010E0:0D 47 5E 13 8E 97 67 C4 BF 9F CE A9 81 CA B7 99 .G^.Ž--gÄ¿ŸÎ©□Ê•™
272. 000010F0:E4 DA CF DB CC BC E9 DA CC E8 DA CC E9 DB CE E5 äÚÏÛÌ¼éÚÌèÚÌéÛÎå
273. 00001100:CF BE CB 73 40 C6 5F 26 C6 5C 22 C3 52 15 CA 75 Ï¾Ës@Æ\_&Æ\"ÃR.Êu
274. 00001110:45 D9 CA B9 E6 D8 CA E3 D5 C6 D1 BF AE C3 AF 9A EÙÊ¹æØÊãÕÆÑ¿®Ã¯š
275. 00001120:CE BE AD D1 C3 B3 CC BD AD D2 C4 B5 D9 CC BD DD Î¾­ÑÃ³Ì½­ÒÄµÙÌ½Ý
276. 00001130:CE BF CE C6 B3 6A 78 3A 47 5C 11 41 5C 0C 41 5C Î¿ÎÆ³jx:G\.A\.A\
277. 00001140:0C 42 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5B 0D .B\.C\.C\.C\.C[.
278. 00001150:42 5A 0C 43 5B 0D 43 5C 0C 42 5B 0B 43 5C 0C 43 BZ.C[.C\.B[.C\.C
279. 00001160:5C 0C 43 5B 0D 43 5B 0D 43 5B 0D 43 5C 0C 43 5C \.C[.C[.C[.C\.C\
280. 00001170:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C .C\.C\.C\.C\.C\.
281. 00001180:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 C\.C\.C\.C\.C\.C
282. 00001190:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 45 5B 0C 45 5B \.C\.C\.C\.E[.E[
283. 000011A0:0C 45 5B 0C 45 5C 0A 00 3F 5F 0E 3F 5F 0E 3F 5F .E[.E\..?\_.?\_.?\_
284. 000011B0:0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 40 5D 0D .?\_.?\_.?\_.?\_.@].
285. 000011C0:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 @].@].@].@].@].@
286. 000011D0:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D ].@].@].@].@].@]
287. 000011E0:0D 41 5E 0E 40 5D 0D 40 5D 0D 40 5D 0D 4C 61 17 .A^.@].@].@].LA.
288. 000011F0:9C A5 7A D1 BF 9E D6 A5 7D E2 BD 9F E3 CF BE DB œ¥zÑ¿žÖ¥}â½ŸãÏ¾Û
289. 00001200:CB BB E9 DA CC E8 D9 CB EA DF D3 DC B0 93 C7 63 Ë»éÚÌèÙËêßÓÜ°"ÇC
290. 00001210:29 C5 5C 22 C5 59 1F C5 5B 21 DD BC A4 DE D2 C4 )Å\"ÅY.Å[!Ý¼¤ÞÒÄ
291. 00001220:E6 D8 CA E8 DA CC CD BB A9 C6 B3 9F CE BD AB CA æØÊèÚÌÍ»©Æ³ŸÎ½«Ê
292. 00001230:B8 A5 CF BE AC D5 C4 B2 BD B4 9D AB AC 8B B6 BA ¸¥Ï¾¬ÕÄ²½´□«¬‹¶º
293. 00001240:9A 78 86 4D 4A 5E 14 43 5C 0C 43 5C 0C 43 5C 0C šx†MJ^.C\.C\.C\.
294. 00001250:43 5C 0C 43 5C 0C 43 5C 0C 43 5B 0D 42 5A 0C 43 C\.C\.C\.C[.BZ.C
295. 00001260:5B 0D 43 5C 0C 42 5B 0B 43 5C 0C 43 5C 0C 43 5B [.C\.B[.C\.C\.C[
296. 00001270:0D 42 5A 0C 43 5B 0D 43 5C 0C 43 5C 0C 43 5C 0C .BZ.C[.C\.C\.C\.
297. 00001280:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 C\.C\.C\.C\.C\.C
298. 00001290:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C \.C\.C\.C\.C\.C\
299. 000012A0:0C 43 5C 0C 43 5C 0C 45 5B 0C 45 5B 0C 45 5B 0C .C\.C\.E[.E[.E[.
300. 000012B0:45 5C 0A 00 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E E\..?\_.?\_.?\_.?\_.
301. 000012C0:3F 5F 0E 3F 5F 0E 3F 5F 0E 40 5D 0D 40 5D 0D 40 ?\_.?\_.?\_.@].@].@
302. 000012D0:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D ].@].@].@].@].@]
303. 000012E0:0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 41 5E 0E .@].@].@].@].A^.
304. 000012F0:40 5D 0D 40 5D 0D 40 5D 0D 53 64 1D A2 AA 82 D7 @].@].@].SD.¢ª‚×
305. 00001300:BC 99 D8 A7 7F DA A8 81 E3 CE BB E1 D2 C3 E8 D9 ¼™Ø§Ú¨□ãÎ»áÒÃèÙ
306. 00001310:CC E8 D9 CB E4 D6 C7 E6 D7 C9 E2 BC A3 D2 84 57 ÌèÙËäÖÇæ×Éâ¼£Ò„W
307. 00001320:C5 56 1A D4 9A 76 E6 DE D2 E0 D0 C1 E3 D5 C6 F0 ÅV.ÔšvæÞÒàÐÁãÕÆð
308. 00001330:E4 D7 E1 D2 C3 CA B7 A4 D1 C1 B0 C9 B7 A3 D0 BD ä×áÒÃÊ•¤ÑÁ°É•£Ð½
309. 00001340:A9 D8 C6 B2 B5 AC 92 7B 89 53 75 89 4D 70 82 46 ©ØÆ²µ¬'{‰Su‰Mp‚F
310. 00001350:53 61 1C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 SA.C\.C\.C\.C\.C
311. 00001360:5C 0C 43 5C 0C 43 5B 0D 42 5A 0C 43 5B 0D 43 5C \.C\.C[.BZ.C[.C\
312. 00001370:0C 42 5B 0B 43 5C 0C 43 5C 0C 43 5B 0D 42 5A 0C .B[.C\.C\.C[.BZ.
313. 00001380:42 5A 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 BZ.C\.C\.C\.C\.C
314. 00001390:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C \.C\.C\.C\.C\.C\
315. 000013A0:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C .C\.C\.C\.C\.C\.
316. 000013B0:43 5C 0C 45 5B 0C 45 5B 0C 45 5B 0C 45 5C 0A 00 C\.E[.E[.E[.E\..
317. 000013C0:3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F ?\_.?\_.?\_.?\_.?\_.?
318. 000013D0:5F 0E 3F 5F 0E 40 5D 0D 40 5D 0D 40 5D 0D 40 5D \_.?\_.@].@].@].@]
319. 000013E0:0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D .@].@].@].@].@].
320. 000013F0:40 5D 0D 40 5D 0D 40 5D 0D 41 5E 0E 40 5D 0D 40 @].@].@].A^.@].@
321. 00001400:5D 0D 41 5D 0E 59 67 24 A9 B2 8C D1 BB 98 D7 A5 ].A].Yg$©²ŒÑ»˜×¥
322. 00001410:7C D4 B0 8D DE D4 C5 DA CA BB DE D1 C5 DB D0 C3 |Ô°□ÞÔÅÚÊ»ÞÑÅÛÐÃ
323. 00001420:DC CE BF E4 D5 C5 EC E2 D7 F0 E5 D9 E4 C4 AE D7 ÜÎ¿äÕÅìâ×ðåÙäÄ®×
324. 00001430:C8 B9 D4 C7 BA D8 C9 BB E0 D2 C4 E9 DD D0 E3 D4 È¹ÔÇºØÉ»àÒÄéÝÐãÔ
325. 00001440:C6 CF BD AB D2 C3 B3 C2 B2 A1 C6 B6 A3 CD BA A7 ÆÏ½«ÒÃ³Â²¡Æ¶£Íº§
326. 00001450:CD BD A7 8B 97 65 7D 8E 55 74 85 49 5A 66 21 44 Í½§‹--E}ŽUt…IZF!D
327. 00001460:5C 0E 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C \.C\.C\.C\.C\.C\
328. 00001470:0C 43 5B 0D 42 5A 0C 43 5B 0D 43 5C 0C 42 5B 0B .C[.BZ.C[.C\.B[.
329. 00001480:43 5C 0C 43 5C 0C 43 5B 0D 42 5A 0C 42 5A 0C 42 C\.C\.C[.BZ.BZ.B
330. 00001490:5B 0B 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C [.C\.C\.C\.C\.C\
331. 000014A0:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C .C\.C\.C\.C\.C\.
332. 000014B0:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 45 C\.C\.C\.C\.C\.E
333. 000014C0:5B 0C 45 5B 0C 45 5B 0C 45 5C 0A 00 3F 5F 0E 3F [.E[.E[.E\..?\_.?
334. 000014D0:5F 0E 3F 5F 0C 3F 5F 0C 3F 5F 0C 3F 5F 0E 40 5E \_.?\_.?\_.?\_.?\_.@^
335. 000014E0:0E 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D .@].@].@].@].@].
336. 000014F0:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 @].@].@].@].@].@
337. 00001500:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 43 5D ].@].@].@].@].C]
338. 00001510:0F 69 71 33 B3 BC 99 CD BA 96 D6 A4 79 A2 9B 72 .iq3³¼™Íº–Ö¤y¢›r
339. 00001520:B3 AE 91 DE D0 C2 D9 C2 AF CE 9D 7C CF 8E 64 CE ³®'ÞÐÂÙÂ¯Î□|ÏŽDÎ
340. 00001530:7D 4C D6 AB 8D DD D4 C9 DB D1 C6 DB CF C2 D8 CA }LÖ«□ÝÔÉÛÑÆÛÏÂØÊ
341. 00001540:BC CF C1 B2 D1 C4 B6 DB CF C3 D6 C8 BB C6 B6 A5 ¼ÏÁ²ÑÄ¶ÛÏÃÖÈ»Æ¶¥
342. 00001550:CF C2 B4 D4 C8 BC A1 9B 81 69 6C 3C 60 63 2E 6C ÏÂ´ÔÈ¼¡›□il<`C.l
343. 00001560:77 41 7A 88 54 74 83 4B 60 6C 27 45 5C 0E 43 5C wAzˆTtƒK`l'E\.C\
344. 00001570:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5B 0D .C\.C\.C\.C\.C[.
345. 00001580:42 5A 0C 43 5B 0D 43 5C 0C 42 5B 0B 43 5C 0C 43 BZ.C[.C\.B[.C\.C
346. 00001590:5C 0C 43 5B 0D 44 5A 0C 44 5A 0C 44 5B 0B 43 5C \.C[.DZ.DZ.D[.C\
347. 000015A0:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C .C\.C\.C\.C\.C\.
348. 000015B0:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 C\.C\.C\.C\.C\.C
349. 000015C0:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5B 0C 43 5B \.C\.C\.C\.C[.C[
350. 000015D0:0C 45 5B 0C 43 5C 0A 00 3F 5F 0E 3F 5F 0E 3F 60 .E[.C\..?\_.?\_.?`
351. 000015E0:0C 3F 60 0C 3F 60 0C 3F 5F 0E 41 5E 0E 40 5D 0D .?`.?`.?\_.A^.@].
352. 000015F0:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 @].@].@].@].@].@
353. 00001600:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D ].@].@].@].@].@]
354. 00001610:0D 40 5D 0D 40 5D 0D 40 5D 0D 45 5E 10 7D 82 48 .@].@].@].E^.}‚H
355. 00001620:BE C5 A4 B9 B0 8A BA B2 93 D1 C9 B6 DB AC 89 E6 ¾Å¤¹°Šº²"ÑÉ¶Û¬‰æ
356. 00001630:CC B7 E8 D2 C0 D1 85 55 CC 75 41 CA 6C 36 C6 6E Ì•èÒÀÑ…UÌuAÊl6Æn
357. 00001640:3B B5 A4 8E D2 C6 B8 E7 D8 C9 E5 D6 C7 E0 D1 C2 ;µ¤ŽÒÆ¸çØÉåÖÇàÑÂ
358. 00001650:DA CC BD DB CD C0 D3 C6 B7 BE AF A0 D0 C3 B5 E9 ÚÌ½ÛÍÀÓÆ•¾¯ ÐÃµé
359. 00001660:DE D2 EA E2 D6 C3 BE A9 8B 89 64 61 66 34 65 6D ÞÒêâÖÃ¾©‹‰DAF4Em
360. 00001670:3E 68 76 45 63 6C 2D 47 5D 0F 43 5C 0C 43 5C 0C >hvECl-G].C\.C\.
361. 00001680:43 5C 0C 43 5C 0C 43 5C 0C 43 5B 0D 42 5A 0C 43 C\.C\.C\.C[.BZ.C
362. 00001690:5B 0D 43 5C 0C 42 5B 0B 43 5C 0C 43 5C 0C 43 5B [.C\.B[.C\.C\.C[
363. 000016A0:0D 45 5B 0D 45 5B 0C 45 5B 0C 43 5C 0C 42 5B 0B .E[.E[.E[.C\.B[.
364. 000016B0:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 C\.C\.C\.C\.C\.C
365. 000016C0:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C \.C\.C\.C\.C\.C\
366. 000016D0:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 45 5B 0C .C\.C\.C\.C\.E[.
367. 000016E0:43 5D 0A 00 3F 5F 0E 3F 5F 0E 3F 60 0C 3F 60 0C C]..?\_.?\_.?`.?`.
368. 000016F0:3F 60 0C 3F 5F 0E 41 5E 0E 40 5D 0D 40 5D 0D 40 ?`.?\_.A^.@].@].@
369. 00001700:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D ].@].@].@].@].@]
370. 00001710:0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D .@].@].@].@].@].
371. 00001720:40 5D 0D 40 5D 0D 47 5E 12 91 93 61 BC C2 A1 C3 @].@].G^.'"A¼Â¡Ã
372. 00001730:C4 AB E8 E0 D4 DF B4 93 D4 96 68 DE B2 93 E0 BB Ä«èàÔß´"Ô–hÞ²"à»
373. 00001740:A0 CF 81 50 CD 7A 48 CB 71 3C C8 63 2A D3 9A 75 Ï□PÍzHËq<ÈC\*Óšu
374. 00001750:E1 D5 C7 DB CB BB D8 C7 B7 D7 C6 B5 D8 C7 B6 D8 áÕÇÛË»ØÇ•×ÆµØÇ¶Ø
375. 00001760:C7 B6 D3 C1 AF C5 B2 9E DB CD BE E9 DE D3 EF E5 Ç¶ÓÁ¯Å²žÛÍ¾éÞÓïå
376. 00001770:DB F6 EE E4 FA F0 E7 E3 D8 CA BB B0 9A A1 9D 83 ÛöîäúðçãØÊ»°š¡□ƒ
377. 00001780:71 74 3B 4A 5E 12 43 5C 0C 43 5C 0C 43 5C 0C 43 qt;J^.C\.C\.C\.C
378. 00001790:5C 0C 43 5C 0C 43 5B 0D 42 5A 0C 43 5B 0D 43 5C \.C\.C[.BZ.C[.C\
379. 000017A0:0C 42 5B 0B 43 5B 0C 43 5B 0C 43 5C 0C 45 5B 0C .B[.C[.C[.C\.E[.
380. 000017B0:44 5A 0B 44 5A 0B 42 5B 0B 42 5B 0B 43 5C 0C 43 DZ.DZ.B[.B[.C\.C
381. 000017C0:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C \.C\.C\.C\.C\.C\
382. 000017D0:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C .C\.C\.C\.C\.C\.
383. 000017E0:43 5C 0C 43 5C 0C 43 5C 0C 45 5B 0C 43 5C 0A 00 C\.C\.C\.E[.C\..
384. 000017F0:3F 5F 0E 3F 5F 0E 3F 60 0C 3F 60 0C 3F 60 0C 3F ?\_.?\_.?`.?`.?`.?
385. 00001800:5F 0E 41 5E 0E 40 5D 0D 40 5D 0D 40 5D 0D 40 5D \_.A^.@].@].@].@]
386. 00001810:0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D .@].@].@].@].@].
387. 00001820:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 @].@].@].@].@].@
388. 00001830:5D 0D 4E 60 17 9B 9C 6F BB C2 9E D7 D5 C2 E0 C0 ].N`.›œo»Âž×ÕÂàÀ
389. 00001840:A6 D7 A0 76 D6 9C 72 D8 A1 7A D4 95 6A D0 82 51 ¦× vÖœrØ¡zÔ•jÐ‚Q
390. 00001850:CD 7A 47 CA 6D 38 D7 A7 88 DB BF AA D9 C9 B8 D3 ÍzGÊm8×§ˆÛ¿ªÙÉ¸Ó
391. 00001860:C3 B2 D0 BF AC D0 BE AC D1 C0 AE CF BD AA C6 B3 Ã²Ð¿¬Ð¾¬ÑÀ®Ï½ªÆ³
392. 00001870:9F D1 BF AD E2 D5 C7 E6 DC D0 EA DF D4 E9 DE D3 ŸÑ¿­âÕÇæÜÐêßÔéÞÓ
393. 00001880:E4 D8 CC DC CE C0 D1 C0 AF CC BD AA 73 79 3E 50 äØÌÜÎÀÑÀ¯Ì½ªsy>P
394. 00001890:5F 17 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C \_.C\.C\.C\.C\.C\
395. 000018A0:0C 43 5B 0D 42 5A 0C 43 5B 0D 43 5C 0C 42 5B 0B .C[.BZ.C[.C\.B[.
396. 000018B0:43 5B 0D 43 5B 0D 43 5C 0C 45 5B 0C 44 5A 0B 44 C[.C[.C\.E[.DZ.D
397. 000018C0:5A 0B 42 5B 0B 42 5B 0B 43 5C 0C 43 5C 0C 43 5C Z.B[.B[.C\.C\.C\
398. 000018D0:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C .C\.C\.C\.C\.C\.
399. 000018E0:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 C\.C\.C\.C\.C\.C
400. 000018F0:5C 0C 43 5C 0C 45 5B 0C 45 5C 0A 00 3F 5F 0E 3F \.C\.E[.E\..?\_.?
401. 00001900:5F 0E 3F 60 0C 3F 60 0C 3F 60 0C 3F 5F 0E 41 5E \_.?`.?`.?`.?\_.A^
402. 00001910:0E 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D .@].@].@].@].@].
403. 00001920:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 @].@].@].@].@].@
404. 00001930:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 57 64 ].@].@].@].@].WD
405. 00001940:1E A5 A5 7C B8 BE 9B D4 D1 BF E2 C5 AD D8 A2 78 .¥¥|¸¾›ÔÑ¿âÅ­Ø¢x
406. 00001950:D7 9E 74 D8 A4 7E D2 8C 5E D0 83 53 CC 74 40 D1 ×žtØ¤~ÒŒ^ÐƒSÌt@Ñ
407. 00001960:8D 62 DE D0 C2 DC CE C0 D4 C2 B1 CF BD AB D1 BF □BÞÐÂÜÎÀÔÂ±Ï½«Ñ¿
408. 00001970:AD D4 C3 B2 D4 C3 B1 CF BE AB D1 BF AD DC CC BC ­ÔÃ²ÔÃ±Ï¾«Ñ¿­ÜÌ¼
409. 00001980:DE D0 C3 DF D3 C6 DD D0 C3 DC CF C1 DF D3 C5 DA ÞÐÃßÓÆÝÐÃÜÏÁßÓÅÚ
410. 00001990:CC BF D3 C5 B6 90 8D 58 60 6D 27 56 64 1E 43 5C Ì¿ÓÅ¶□□X`m'VD.C\
411. 000019A0:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5B 0D .C\.C\.C\.C\.C[.
412. 000019B0:42 5A 0C 43 5B 0D 43 5C 0C 42 5B 0B 43 5B 0D 43 BZ.C[.C\.B[.C[.C
413. 000019C0:5B 0D 43 5C 0C 44 5A 0B 44 5A 0B 44 5A 0B 42 5B [.C\.DZ.DZ.DZ.B[
414. 000019D0:0B 42 5B 0B 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C .B[.C\.C\.C\.C\.
415. 000019E0:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 C\.C\.C\.C\.C\.C
416. 000019F0:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C \.C\.C\.C\.C\.C\
417. 00001A00:0C 45 5B 0C 45 5C 0A 00 3F 5F 0E 3F 5F 0E 3F 60 .E[.E\..?\_.?\_.?`
418. 00001A10:0C 3F 60 0C 3F 60 0C 3F 5F 0E 41 5E 0E 40 5D 0D .?`.?`.?\_.A^.@].
419. 00001A20:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 @].@].@].@].@].@
420. 00001A30:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D ].@].@].@].@].@]
421. 00001A40:0D 40 5D 0D 40 5D 0D 40 5D 0D 62 6B 26 B0 B1 8C .@].@].@].Bk&°±Œ
422. 00001A50:BB BE A0 E3 DD D0 E8 D7 C8 D8 A2 7A D7 A4 7D D5 »¾ ãÝÐè×ÈØ¢z×¤}Õ
423. 00001A60:9C 74 D1 87 57 CE 7C 4A D0 88 5C D7 C2 AF D4 C5 œtÑ‡WÎ|JÐˆ\×Â¯ÔÅ
424. 00001A70:B5 D5 C4 B2 D9 C9 B8 DD CD BD DC CC BC DA C9 B9 µÕÄ²ÙÉ¸ÝÍ½ÜÌ¼ÚÉ¹
425. 00001A80:D7 C6 B6 D7 C6 B5 DF D0 C0 D7 C7 B6 D7 C8 B9 D5 ×Æ¶×ÆµßÐÀ×Ç¶×È¹Õ
426. 00001A90:C7 B9 D9 CB BD E5 DA CD E4 D9 CC E0 D3 C6 D7 CD Ç¹ÙË½åÚÍäÙÌàÓÆ×Í
427. 00001AA0:BD 7C 84 48 5B 6A 1F 5C 67 21 43 5C 0C 43 5C 0C ½|„H[j.\g!C\.C\.
428. 00001AB0:43 5C 0C 43 5C 0C 43 5C 0C 43 5B 0D 42 5A 0C 43 C\.C\.C\.C[.BZ.C
429. 00001AC0:5B 0D 43 5C 0C 42 5B 0B 43 5B 0D 42 5A 0C 42 5B [.C\.B[.C[.BZ.B[
430. 00001AD0:0B 44 5A 0B 44 5A 0B 44 5A 0B 42 5B 0B 42 5B 0B .DZ.DZ.DZ.B[.B[.
431. 00001AE0:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 C\.C\.C\.C\.C\.C
432. 00001AF0:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C \.C\.C\.C\.C\.C\
433. 00001B00:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 45 5B 0C .C\.C\.C\.C\.E[.
434. 00001B10:45 5C 0A 00 3F 5F 0E 3F 5F 0E 3F 60 0C 3F 60 0C E\..?\_.?\_.?`.?`.
435. 00001B20:3F 60 0C 3F 5F 0E 41 5E 0E 40 5D 0D 40 5D 0D 40 ?`.?\_.A^.@].@].@
436. 00001B30:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D ].@].@].@].@].@]
437. 00001B40:0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D .@].@].@].@].@].
438. 00001B50:40 5D 0D 41 5D 0D 6D 73 32 B9 BA 9B D6 D1 BF EC @].A].ms2¹º›ÖÑ¿ì
439. 00001B60:E3 DA EE E7 DF E1 C0 A5 D7 AB 88 D7 A3 7E DA A7 ãÚîçßáÀ¥×«ˆ×£~Ú§
440. 00001B70:84 DD B1 94 D9 C4 B1 D4 C5 B4 DB CB BB E9 DB CD „Ý±"ÙÄ±ÔÅ´ÛË»éÛÍ
441. 00001B80:E9 DB CE E3 D5 C6 DE CF BF DC CD BC DF CF C0 E1 éÛÎãÕÆÞÏ¿ÜÍ¼ßÏÀá
442. 00001B90:D1 C2 DF CF C0 DA CA B9 D6 C7 B8 D5 C7 B9 E8 DC ÑÂßÏÀÚÊ¹ÖÇ¸ÕÇ¹èÜ
443. 00001BA0:D1 E8 DC D1 E4 D8 CC E5 D9 CC B2 AD 93 7D 89 53 ÑèÜÑäØÌåÙÌ²­"}‰S.
444. 00001BB0:6A 77 32 60 6A 24 44 5C 0C 43 5C 0C 43 5C 0C 43 jw2`j$D\.C\.C\.C
445. 00001BC0:5C 0C 43 5C 0C 43 5B 0D 42 5A 0C 43 5B 0D 43 5C \.C\.C[.BZ.C[.C\
446. 00001BD0:0C 42 5B 0B 42 5A 0C 42 5A 0C 42 5B 0B 44 5A 0B .B[.BZ.BZ.B[.DZ.
447. 00001BE0:44 5A 0B 44 5A 0B 42 5B 0B 42 5B 0B 43 5C 0C 43 DZ.DZ.B[.B[.C\.C
448. 00001BF0:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C \.C\.C\.C\.C\.C\
449. 00001C00:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C .C\.C\.C\.C\.C\.
450. 00001C10:43 5C 0C 43 5C 0C 43 5C 0C 45 5B 0C 45 5C 0A 00 C\.C\.C\.E[.E\..
451. 00001C20:40 5D 0D 41 5E 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F @].A^.?\_.?\_.?\_.?
452. 00001C30:5F 0E 41 5E 0E 40 5D 0D 40 5D 0D 40 5D 0D 40 5D \_.A^.@].@].@].@]
453. 00001C40:0D 41 5E 0E 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D .A^.@].@].@].@].
454. 00001C50:41 5E 0E 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 41 A^.@].@].@].@].A
455. 00001C60:5E 0F 77 79 3C CC CA B4 EB E2 D9 ED E4 DB EF E7 ^.wy<ÌÊ´ëâÙíäÛïç
456. 00001C70:E0 EF E7 DE DE CC BB DD CD BC E8 DB CD E9 DC CF àïçÞÞÌ»ÝÍ¼èÛÍéÜÏ
457. 00001C80:DC CD BD E5 D6 C7 F1 E4 D8 F0 E3 D6 EA DC CE E5 ÜÍ½åÖÇñäØðãÖêÜÎå
458. 00001C90:D6 C7 E1 D2 C2 DF CF C0 DF CF C0 DD CD BD D9 C8 ÖÇáÒÂßÏÀßÏÀÝÍ½ÙÈ
459. 00001CA0:B8 D7 C6 B5 E0 D3 C6 E6 DB CF EB E0 D5 E1 D5 C8 ¸×ÆµàÓÆæÛÏëàÕáÕÈ
460. 00001CB0:DF D3 C7 E4 D8 CB A7 A3 88 77 82 4F 77 86 4B 6B ßÓÇäØË§£ˆw‚Ow†Kk
461. 00001CC0:71 30 44 5D 0E 41 5C 0C 43 5C 0C 43 5C 0C 43 5C q0D].A\.C\.C\.C\
462. 00001CD0:0C 43 5C 0C 43 5C 0C 43 5B 0D 43 5B 0D 43 5B 0D .C\.C\.C[.C[.C[.
463. 00001CE0:42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 44 5A 0B 44 B[.B[.B[.B[.DZ.D
464. 00001CF0:5A 0B 42 5B 0B 42 5B 0B 44 5A 0B 43 5C 0C 43 5C Z.B[.B[.DZ.C\.C\
465. 00001D00:0C 43 5D 0A 43 5D 0A 43 5D 0A 43 5C 0C 43 5C 0C .C].C].C].C\.C\.
466. 00001D10:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 C\.C\.C\.C\.C\.C
467. 00001D20:5C 0C 43 5C 0C 42 5B 0B 44 5B 09 00 40 5D 0D 41 \.C\.B[.D[..@].A
468. 00001D30:5E 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 41 5E ^.?\_.?\_.?\_.?\_.A^
469. 00001D40:0E 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 41 5E 0E .@].@].@].@].A^.
470. 00001D50:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 @].@].@].@].@].@
471. 00001D60:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 47 5F 13 80 7F ].@].@].@].G\_.€
472. 00001D70:46 DE DA C9 EF E6 DD EE E5 DD F0 E8 E0 F1 E8 E0 FÞÚÉïæÝîåÝðèàñèà
473. 00001D80:E2 D2 C3 DE CD BD E3 D3 C3 E3 D4 C5 E2 D3 C5 E6 âÒÃÞÍ½ãÓÃãÔÅâÓÅæ
474. 00001D90:D9 CA E5 D7 C9 E7 DA CB E9 DA CC E5 D6 C8 E2 D3 ÙÊå×ÉçÚËéÚÌåÖÈâÓ
475. 00001DA0:C4 E0 D1 C2 DA CA B9 D3 C2 B1 CF BD AA CC B9 A6 ÄàÑÂÚÊ¹ÓÂ±Ï½ªÌ¹¦
476. 00001DB0:D6 C7 B8 E1 D5 C9 E7 DB CF DF D3 C6 E0 D3 C7 DE ÖÇ¸áÕÉçÛÏßÓÆàÓÇÞ
477. 00001DC0:D1 C5 D4 C9 B9 7F 85 5B 75 83 4A 74 78 3A 4A 5F ÑÅÔÉ¹…[uƒJtx:J\_
478. 00001DD0:14 41 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C .A\.C\.C\.C\.C\.
479. 00001DE0:42 5B 0B 42 5B 0C 42 5B 0C 43 5B 0C 42 5B 0B 42 B[.B[.B[.C[.B[.B
480. 00001DF0:5B 0B 42 5B 0B 42 5B 0B 43 5A 0B 43 5A 0B 42 5B [.B[.B[.CZ.CZ.B[
481. 00001E00:0B 42 5B 0B 44 5A 0B 43 5C 0C 43 5C 0C 43 5D 0A .B[.DZ.C\.C\.C].
482. 00001E10:43 5D 0A 43 5D 0A 43 5C 0C 43 5C 0C 43 5C 0C 43 C].C].C\.C\.C\.C
483. 00001E20:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C \.C\.C\.C\.C\.C\
484. 00001E30:0C 42 5B 0B 44 5B 09 00 40 5D 0D 41 5E 0E 3F 5F .B[.D[..@].A^.?\_
485. 00001E40:0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 41 5E 0E 40 5D 0D .?\_.?\_.?\_.A^.@].
486. 00001E50:40 5D 0D 40 5D 0D 40 5D 0D 41 5E 0E 40 5D 0D 40 @].@].@].A^.@].@
487. 00001E60:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D ].@].@].@].@].@]
488. 00001E70:0D 40 5D 0D 40 5D 0D 4B 61 15 8E 8B 57 CF CE B6 .@].@].KA.Ž‹WÏÎ¶
489. 00001E80:EA E2 D6 F0 E7 DF F1 E9 E1 F1 E8 E0 E4 D5 C5 E1 êâÖðçßñéáñèàäÕÅá
490. 00001E90:D1 C0 E0 D0 C0 E0 D0 C2 E4 D5 C6 E3 D4 C5 E3 D4 ÑÀàÐÀàÐÂäÕÆãÔÅãÔ
491. 00001EA0:C5 E5 D6 C8 E8 DB CD EB DD D0 EB DD CF E8 DA CC ÅåÖÈèÛÍëÝÐëÝÏèÚÌ
492. 00001EB0:E0 D1 C2 D7 C6 B5 CE BB A9 C9 B5 A2 D3 C4 B5 D3 àÑÂ×ÆµÎ»©Éµ¢ÓÄµÓ
493. 00001EC0:C4 B6 DA CD BF E1 D5 C9 E3 D7 CB DD D1 C4 EA DF Ä¶ÚÍ¿áÕÉã×ËÝÑÄêß
494. 00001ED0:D4 B7 B4 9C 77 85 4D 76 7D 3E 4F 61 17 41 5C 0C Ô•´œw…Mv}>OA.A\.
495. 00001EE0:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 42 5B 0B 42 C\.C\.C\.C\.B[.B
496. 00001EF0:5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B [.B[.B[.B[.B[.B[
497. 00001F00:0B 42 5B 0B 42 5B 0B 42 5B 0B 43 5A 0B 43 5A 0B .B[.B[.B[.CZ.CZ.
498. 00001F10:44 5A 0B 43 5C 0C 43 5C 0C 43 5D 0A 43 5D 0A 43 DZ.C\.C\.C].C].C
499. 00001F20:5D 0A 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C ].C\.C\.C\.C\.C\
500. 00001F30:0C 43 5C 0C 43 5C 0C 43 5C 0C 42 5B 0B 43 5A 0B .C\.C\.C\.B[.CZ.
501. 00001F40:44 5B 09 00 40 5D 0D 40 5D 0D 3F 5F 0E 3F 5F 0E D[..@].@].?\_.?\_.
502. 00001F50:3F 5F 0E 3F 5F 0E 40 5D 0D 40 5D 0D 40 5D 0D 40 ?\_.?\_.@].@].@].@
503. 00001F60:5D 0D 40 5D 0D 41 5E 0E 40 5D 0D 40 5D 0D 40 5D ].@].A^.@].@].@]
504. 00001F70:0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D .@].@].@].@].@].
505. 00001F80:40 5D 0D 50 63 1A A3 9F 71 CA C9 B1 E0 D9 CB F1 @].PC.£ŸqÊÉ±àÙËñ
506. 00001F90:E8 E0 F2 EA E2 F2 E9 E0 E7 D8 C8 E4 D4 C4 E1 D0 èàòêâòéàçØÈäÔÄáÐ
507. 00001FA0:C1 E4 D5 C6 DD CD BD D7 C6 B6 DE CF BF E5 D6 C8 ÁäÕÆÝÍ½×Æ¶ÞÏ¿åÖÈ
508. 00001FB0:EB DE D0 F0 E3 D6 F1 E5 D8 F0 E3 D6 ED DF D3 E9 ëÞÐðãÖñåØðãÖíßÓé
509. 00001FC0:DB CD E8 DA CC E3 D4 C5 E3 D7 CA E4 D8 CC D9 CB ÛÍèÚÌãÔÅã×ÊäØÌÙË
510. 00001FD0:BD D9 CB BD E0 D3 C7 DF D3 C6 E7 DC D0 E1 D8 C8 ½ÙË½àÓÇßÓÆçÜÐáØÈ
511. 00001FE0:80 8D 56 7A 82 45 55 64 1B 41 5C 0C 43 5C 0C 43 €□Vz‚EUD.A\.C\.C
512. 00001FF0:5C 0C 43 5C 0C 42 5B 0B 42 5B 0B 42 5B 0B 42 5B \.C\.B[.B[.B[.B[
513. 00002000:0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B .B[.B[.B[.B[.B[.
514. 00002010:42 5B 0B 42 5B 0B 44 5A 0B 44 5A 0B 44 5A 0B 43 B[.B[.DZ.DZ.DZ.C
515. 00002020:5C 0C 43 5C 0C 43 5D 0A 43 5D 0A 43 5D 0A 43 5C \.C\.C].C].C].C\
516. 00002030:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C .C\.C\.C\.C\.C\.
517. 00002040:43 5C 0C 43 5C 0C 43 5A 0B 44 5A 0B 44 5B 09 00 C\.C\.CZ.DZ.D[..
518. 00002050:40 5D 0D 40 5D 0D 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F @].@].?\_.?\_.?\_.?
519. 00002060:5F 0E 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D \_.@].@].@].@].@]
520. 00002070:0D 41 5E 0E 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D .A^.@].@].@].@].
521. 00002080:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 58 @].@].@].@].@].X
522. 00002090:67 20 B2 AF 88 D8 D4 C5 F0 E7 DE F1 E9 E1 F3 EC g ²¯ˆØÔÅðçÞñéáóì
523. 000020A0:E4 F3 E9 E1 E9 DA CB E6 D6 C7 E4 D4 C4 E6 D8 C9 äóéáéÚËæÖÇäÔÄæØÉ
524. 000020B0:DF D0 C0 D5 C4 B3 DB CB BA E5 D7 C9 EB DD D0 EF ßÐÀÕÄ³ÛËºå×ÉëÝÐï
525. 000020C0:E2 D5 F1 E4 D8 F0 E4 D7 ED DF D2 EE E1 D4 E6 D8 âÕñäØðä×íßÒîáÔæØ
526. 000020D0:C9 C1 AE 98 CA B9 A8 E8 DC D1 EC E1 D7 E7 DC D0 ÉÁ®˜Ê¹¨èÜÑìá×çÜÐ
527. 000020E0:DD D0 C3 D4 C6 B7 D8 CB BC AE AC 8B 79 87 4D 7B ÝÐÃÔÆ•ØË¼®¬‹y‡M{
528. 000020F0:84 4A 5C 67 20 41 5C 0C 43 5C 0C 43 5C 0C 43 5C „J\g A\.C\.C\.C\
529. 00002100:0C 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B .B[.B[.B[.B[.B[.
530. 00002110:42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 B[.B[.B[.B[.B[.B
531. 00002120:5B 0B 44 5A 0B 44 5A 0B 44 5A 0B 43 5C 0C 43 5C [.DZ.DZ.DZ.C\.C\
532. 00002130:0C 43 5D 0A 43 5D 0A 43 5D 0A 43 5C 0C 43 5C 0C .C].C].C].C\.C\.
533. 00002140:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 42 C\.C\.C\.C\.C\.B
534. 00002150:5B 0B 44 5A 0B 44 5A 0B 44 5B 09 00 40 5D 0D 40 [.DZ.DZ.D[..@].@
535. 00002160:5D 0D 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 40 5D ].?\_.?\_.?\_.?\_.@]
536. 00002170:0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 41 5E 0E .@].@].@].@].A^.
537. 00002180:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 @].@].@].@].@].@
538. 00002190:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 64 6C 29 C3 BD ].@].@].@].Dl)Ã½
539. 000021A0:A0 F4 EC E3 F2 E9 E0 F3 EA E2 F5 EE E6 F3 EA E1 ôìãòéàóêâõîæóêá
540. 000021B0:EB DC CD E8 D9 CA E5 D5 C6 E4 D4 C5 E3 D3 C3 D9 ëÜÍèÙÊåÕÆäÔÅãÓÃÙ
541. 000021C0:C8 B8 D3 C1 AF DC CC BC EB DD D0 EF E2 D6 F2 E5 È¸ÓÁ¯ÜÌ¼ëÝÐïâÖòå
542. 000021D0:DA F2 E5 D9 F4 E7 DB EA DC CE C7 B3 A0 B7 A2 8B ÚòåÙôçÛêÜÎÇ³ •¢‹
543. 000021E0:D1 C2 B3 EB E0 D6 E6 DA CF EA E0 D5 EF E4 DB E8 ÑÂ³ëàÖæÚÏêàÕïäÛè
544. 000021F0:DC D1 D0 C5 B4 8A 8F 63 7C 8B 51 79 85 4A 64 69 ÜÑÐÅ´Š□C|‹Qy…JDi
545. 00002200:27 41 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 42 5B 0B 'A\.C\.C\.C\.B[.
546. 00002210:42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 B[.B[.B[.B[.B[.B
547. 00002220:5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 44 5A [.B[.B[.B[.B[.DZ
548. 00002230:0B 44 5A 0B 44 5A 0B 43 5C 0C 43 5C 0C 43 5D 0A .DZ.DZ.C\.C\.C].
549. 00002240:43 5D 0A 43 5D 0A 43 5C 0C 43 5C 0C 43 5C 0C 43 C].C].C\.C\.C\.C
550. 00002250:5C 0C 43 5C 0C 43 5C 0C 42 5B 0B 44 5A 0B 44 5A \.C\.C\.B[.DZ.DZ
551. 00002260:0B 44 5A 0B 44 5B 09 00 40 5D 0D 40 5D 0D 3F 5E .DZ.D[..@].@].?^
552. 00002270:0D 3F 5E 0D 3F 5E 0D 3F 5E 0D 40 5D 0D 40 5D 0D .?^.?^.?^.@].@].
553. 00002280:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 @].@].@].@].@].@
554. 00002290:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D ].@].@].@].@].@]
555. 000022A0:0D 3F 5C 0C 40 5C 0D 72 74 35 CC C5 AC F4 EC E4 .?\.@\.rt5ÌÅ¬ôìä
556. 000022B0:F1 E9 E0 F6 EE E6 F5 EE E6 F2 E9 DE ED DE CF EB ñéàöîæõîæòéÞíÞÏë
557. 000022C0:DC CE E3 D3 C3 D8 C7 B4 DB CB BA DF D0 BF D9 C8 ÜÎãÓÃØÇ´ÛËºßÐ¿ÙÈ
558. 000022D0:B7 D9 C8 B7 ED DE D0 F4 E6 D9 F9 ED E0 FD F0 E4 •ÙÈ•íÞÐôæÙùíàýðä
559. 000022E0:FF F3 E7 E0 CF BD CA B6 A1 C7 B3 9D F3 E7 DC F4 ÿóçàÏ½Ê¶¡Ç³□óçÜô
560. 000022F0:EA E0 E7 DD D0 E7 DC D0 E8 DD D2 EF E5 DB F0 E7 êàçÝÐçÜÐèÝÒïåÛðç
561. 00002300:DD A5 A6 81 7C 8A 50 76 83 47 6F 71 31 43 5C 0E Ý¥¦□|ŠPvƒGoq1C\.
562. 00002310:43 5C 0C 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 C\.B[.B[.B[.B[.B
563. 00002320:5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B [.B[.B[.B[.B[.B[
564. 00002330:0B 42 5B 0B 42 5B 0B 42 5B 0B 44 5A 0B 44 5A 0B .B[.B[.B[.DZ.DZ.
565. 00002340:43 5A 0B 42 5B 0B 42 5B 0B 43 5C 0A 43 5C 0A 43 CZ.B[.B[.C\.C\.C
566. 00002350:5C 0A 43 5C 0C 43 5C 0C 43 5C 0B 43 5C 0B 43 5C \.C\.C\.C\.C\.C\
567. 00002360:0C 43 5B 0C 42 5A 0A 43 59 0A 44 5A 0B 44 5A 0B .C[.BZ.CY.DZ.DZ.
568. 00002370:43 5B 09 00 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D C[..@].@].@].@].
569. 00002380:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 @].@].@].@].@].@
570. 00002390:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D ].@].@].@].@].@]
571. 000023A0:0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 3F 5C 0C .@].@].@].@].?\.
572. 000023B0:43 5E 0F 80 7E 41 D3 CD BA F2 E9 E0 F4 EB E2 F4 C^.€~AÓÍºòéàôëâô
573. 000023C0:EA E1 F0 E8 DF E5 D9 CD DD CE BE E4 D5 C7 DC CD êáðèßåÙÍÝÎ¾äÕÇÜÍ
574. 000023D0:BF D9 CA BB D9 CA BB DA CC BD D2 C4 B6 CC BF B2 ¿ÙÊ»ÙÊ»ÚÌ½ÒÄ¶Ì¿²
575. 000023E0:CC C1 B5 C8 BD B2 C3 B9 AF C0 B6 AB BA AE A2 AA ÌÁµÈ½²Ã¹¯À¶«º®¢ª
576. 000023F0:9A 8B A6 95 84 BB AD 9E D7 CE C5 D5 CA BF D3 C6 š‹¦•„»­ž×ÎÅÕÊ¿ÓÆ
577. 00002400:B9 DC CF C1 E5 D7 C9 EB DF D2 E6 DC CF 8D 92 67 ¹ÜÏÁå×ÉëßÒæÜÏ□'g
578. 00002410:7A 87 4D 73 82 45 79 79 39 49 5E 11 43 5C 0C 42 z‡Ms‚Eyy9I^.C\.B
579. 00002420:5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B [.B[.B[.B[.B[.B[
580. 00002430:0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B .B[.B[.B[.B[.B[.
581. 00002440:42 5B 0B 42 5B 0B 44 5A 0B 44 5A 0B 42 5B 0B 42 B[.B[.DZ.DZ.B[.B
582. 00002450:5B 0B 42 5B 0B 43 5C 0C 43 5C 0C 43 5C 0C 43 5C [.B[.C\.C\.C\.C\
583. 00002460:0C 43 5C 0C 43 5D 0A 43 5D 0A 43 5C 0C 44 5A 0B .C\.C].C].C\.DZ.
584. 00002470:43 59 0A 43 59 0A 44 5A 0B 44 5A 0B 42 5C 09 00 CY.CY.DZ.DZ.B\..
585. 00002480:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 @].@].@].@].@].@
586. 00002490:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D ].@].@].@].@].@]
587. 000024A0:0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D .@].@].@].@].@].
588. 000024B0:40 5D 0D 40 5D 0D 40 5D 0D 3F 5C 0C 45 5F 11 90 @].@].@].?\.E\_.□
589. 000024C0:8A 4F EA E1 D5 E8 DF D7 E7 DF D6 E7 E1 DA E5 E0 ŠOêáÕèß×çßÖçáÚåà
590. 000024D0:DB E6 E3 DF EA E8 E5 F0 EE ED F2 F2 F1 F1 F1 F1 Ûæãßêèåðîíòòññññ
591. 000024E0:F0 F0 F0 EB EB EB E3 E4 E4 D7 D7 D8 C2 C3 C4 A8 ðððëëëãää××ØÂÃÄ¨
592. 000024F0:A9 AA 8B 8B 8C 6C 6C 6D 53 54 55 45 45 46 3E 3F ©ª‹‹ŒllmSTUEEF>?
593. 00002500:3F 3B 3B 3B 3C 3B 3A 41 40 3F 4A 48 46 5B 57 54 ?;;;<;:A@?JHF[WT
594. 00002510:78 72 6C 9C 94 8B B4 AB A2 A2 9F 86 95 9C 6F 7A xrlœ"‹´«¢¢Ÿ†•œoz
595. 00002520:8A 4F 7B 7D 3D 4B 5F 13 43 5C 0C 42 5B 0B 42 5B ŠO{}=K\_.C\.B[.B[
596. 00002530:0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B .B[.B[.B[.B[.B[.
597. 00002540:42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 B[.B[.B[.B[.B[.B
598. 00002550:5B 0B 44 5A 0B 44 5A 0B 42 5B 0B 42 5B 0B 42 5B [.DZ.DZ.B[.B[.B[
599. 00002560:0B 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C .C\.C\.C\.C\.C\.
600. 00002570:43 5D 0A 43 5D 0A 43 5C 0C 44 5A 0B 43 59 0A 43 C].C].C\.DZ.CY.C
601. 00002580:59 0A 44 5A 0B 44 5A 0B 42 5C 09 00 40 5D 0D 40 Y.DZ.DZ.B\..@].@
602. 00002590:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D ].@].@].@].@].@]
603. 000025A0:0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D .@].@].@].@].@].
604. 000025B0:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 @].@].@].@].@].@
605. 000025C0:5D 0D 40 5D 0D 3F 5C 0C 52 64 1D A1 92 80 C5 C5 ].@].?\.RD.¡'€ÅÅ
606. 000025D0:C4 D2 D2 D2 E1 E2 E3 EB EC ED F2 F2 F3 F5 F5 F5 ÄÒÒÒáâãëìíòòóõõõ
607. 000025E0:F7 F7 F6 F6 F6 F6 F5 F4 F3 F4 F3 F1 F1 EF EE F1 ÷÷ööööõôóôóññïîñ
608. 000025F0:EF ED ED EC EA E7 E5 E3 DF DE DC D2 D1 D0 C4 C2 ïííìêçåãßÞÜÒÑÐÄÂ
609. 00002600:C0 B7 B5 B2 AB A9 A6 A0 9D 9B 93 90 8D 89 88 87 À•µ²«©¦ □›"□□‰ˆ‡
610. 00002610:7E 7E 7F 72 72 71 6B 6A 6A 69 69 6A 69 6A 6B 6F ~~rrqkjjiijijko
611. 00002620:71 73 83 86 89 A2 A3 A5 BB BA BA C4 C7 BF AE A4 qsƒ†‰¢£¥»ººÄÇ¿®¤
612. 00002630:81 55 64 1C 43 5C 0C 42 5B 0B 42 5B 0B 42 5B 0B □UD.C\.B[.B[.B[.
613. 00002640:42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 B[.B[.B[.B[.B[.B
614. 00002650:5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 44 5A [.B[.B[.B[.B[.DZ
615. 00002660:0B 44 5A 0B 42 5B 0B 42 5B 0B 42 5B 0B 43 5C 0C .DZ.B[.B[.B[.C\.
616. 00002670:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5D 0A 43 C\.C\.C\.C\.C].C
617. 00002680:5D 0A 43 5C 0C 44 5A 0B 44 5A 0B 43 59 0A 43 59 ].C\.DZ.DZ.CY.CY
618. 00002690:0A 44 5A 0B 42 5C 09 00 40 5D 0D 40 5D 0D 40 5D .DZ.B\..@].@].@]
619. 000026A0:0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D .@].@].@].@].@].
620. 000026B0:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 @].@].@].@].@].@
621. 000026C0:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D ].@].@].@].@].@]
622. 000026D0:0D 3F 5C 0C 75 82 5E 9E A0 A2 C6 C7 C8 DE DE DE .?\.u‚^ž ¢ÆÇÈÞÞÞ
623. 000026E0:EA E9 E8 F2 F1 F0 F5 F3 F0 F5 F3 F0 F1 ED E9 EB êéèòñðõóðõóðñíéë
624. 000026F0:E6 E0 E9 E3 DD E6 E1 DA E0 DB D6 E9 E5 DD F6 F4 æàéãÝæáÚàÛÖéåÝöô
625. 00002700:F3 E9 E5 E1 E4 DF DA DC D7 D0 D2 CA C1 D4 CC C4 óéåáäßÚÜ×ÐÒÊÁÔÌÄ
626. 00002710:DB D4 CC E2 DC D5 D4 CC C3 CF C7 BC F3 EF EB DF ÛÔÌâÜÕÔÌÃÏÇ¼óïëß
627. 00002720:D9 D2 D9 D2 CB DC D7 D1 DB D8 D4 C9 C4 B8 B8 AF ÙÒÙÒËÜ×ÑÛØÔÉÄ¸¸¯
628. 00002730:96 C3 BF B7 D0 D0 D0 D0 D1 D2 CF D1 D3 86 8F 6D –Ã¿•ÐÐÐÐÑÒÏÑÓ†□m
629. 00002740:43 5C 0C 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 C\.B[.B[.B[.B[.B
630. 00002750:5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B [.B[.B[.B[.B[.B[
631. 00002760:0B 42 5B 0B 42 5B 0B 42 5B 0B 44 5A 0B 44 5A 0B .B[.B[.B[.DZ.DZ.
632. 00002770:42 5B 0B 42 5B 0B 42 5B 0B 43 5C 0C 43 5C 0C 43 B[.B[.B[.C\.C\.C
633. 00002780:5C 0C 43 5C 0C 43 5C 0C 43 5D 0A 43 5D 0A 44 5A \.C\.C\.C].C].DZ
634. 00002790:0B 44 5A 0B 44 5A 0B 43 59 0A 43 59 0A 44 5A 0B .DZ.DZ.CY.CY.DZ.
635. 000027A0:42 5C 09 00 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D B\..@].@].@].@].
636. 000027B0:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 @].@].@].@].@].@
637. 000027C0:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D ].@].@].@].@].@]
638. 000027D0:0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 3F 5C 0C .@].@].@].@].?\.
639. 000027E0:89 91 7B AD AC AC AD A4 84 A2 97 5E A7 9B 5F EE ‰'{­¬¬­¤„¢--^§›\_î
640. 000027F0:E9 DA FC FC FC F8 F6 F4 EE E9 E4 E5 DE D7 E6 E1 éÚüüüøöôîéäåÞ×æá
641. 00002800:DB E4 E0 DB B8 AA 7D B7 AD 7C FF FF FF F2 F2 F2 ÛäàÛ¸ª}•­|ÿÿÿòòò
642. 00002810:E5 E1 DD DE D9 D3 D6 CF C8 DB D6 D0 DB D6 D0 D9 åáÝÞÙÓÖÏÈÛÖÐÛÖÐÙ
643. 00002820:D3 CD D1 CA C2 BE B4 A8 BA AF A3 C6 BD B3 CE C6 ÓÍÑÊÂ¾´¨º¯£Æ½³ÎÆ
644. 00002830:BE D1 CD C7 C5 B8 99 9A 8F 4D 91 88 43 9A 90 4E ¾ÑÍÇÅ¸™š□M'ˆCš□N
645. 00002840:A5 99 61 B2 A9 8B CB CB CC 9D A4 90 43 5C 0C 42 ¥™A²©‹ËËÌ□¤□C\.B
646. 00002850:5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B [.B[.B[.B[.B[.B[
647. 00002860:0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B .B[.B[.B[.B[.B[.
648. 00002870:42 5B 0B 42 5B 0B 44 5A 0B 44 5A 0B 42 5B 0B 42 B[.B[.DZ.DZ.B[.B
649. 00002880:5B 0B 42 5B 0B 43 5C 0C 43 5C 0C 43 5C 0C 43 5C [.B[.C\.C\.C\.C\
650. 00002890:0C 43 5C 0C 43 5D 0A 43 5D 0A 44 5A 0B 44 5A 0B .C\.C].C].DZ.DZ.
651. 000028A0:43 59 0A 43 59 0A 43 59 0A 44 5A 0B 42 5C 09 00 CY.CY.CY.DZ.B\..
652. 000028B0:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 @].@].@].@].@].@
653. 000028C0:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D ].@].@].@].@].@]
654. 000028D0:0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D .@].@].@].@].@].
655. 000028E0:40 5D 0D 40 5D 0D 40 5D 0D 3F 5C 0C 90 99 7F 7D @].@].@].?\.□™}
656. 000028F0:7C 7C 73 70 69 83 7D 64 93 88 5F A6 9C 78 EA E8 ||spiƒ}D"ˆ\_¦œxêè
657. 00002900:E4 FB F9 F7 E7 E1 DA E1 D9 D1 D9 D0 BF AB 9F 68 äûù÷çáÚáÙÑÙÐ¿«Ÿh
658. 00002910:8D 84 3E 99 8F 4C C8 BB 90 E6 E0 CA F3 F0 EB E9 □„>™□LÈ»□æàÊóðëé
659. 00002920:E5 E1 E0 DA D4 E6 E2 DE DD D7 D2 CE C7 BF DF DA åáàÚÔæâÞÝ×ÒÎÇ¿ßÚ
660. 00002930:D5 E1 DC D7 D5 CE C6 D6 CF C7 CC C3 B4 BC B1 99 ÕáÜ×ÕÎÆÖÏÇÌÃ´¼±™
661. 00002940:96 8A 54 8C 80 47 90 84 56 8E 83 5D 89 82 6C 9A –ŠTŒ€G□„VŽƒ]‰‚lš
662. 00002950:97 91 B2 B2 B3 92 9B 80 43 5C 0C 42 5B 0B 42 5B --'²²³'›€C\.B[.B[
663. 00002960:0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B .B[.B[.B[.B[.B[.
664. 00002970:42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 B[.B[.B[.B[.B[.B
665. 00002980:5B 0B 44 5A 0B 44 5A 0B 42 5B 0B 42 5B 0B 42 5B [.DZ.DZ.B[.B[.B[
666. 00002990:0B 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C .C\.C\.C\.C\.C\.
667. 000029A0:43 5D 0A 42 5C 09 44 5A 0B 44 5A 0B 43 59 0A 43 C].B\.DZ.DZ.CY.C
668. 000029B0:59 0A 44 5A 0B 44 5A 0B 42 5C 09 00 40 5D 0B 40 Y.DZ.DZ.B\..@].@
669. 000029C0:5D 0B 40 5E 0C 40 5E 0C 40 5E 0C 40 5E 0C 40 5E ].@^.@^.@^.@^.@^
670. 000029D0:0C 40 5D 0B 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D .@].@].@].@].@].
671. 000029E0:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 @].@].@].@].@].@
672. 000029F0:5D 0D 40 5D 0D 3F 5C 0C 49 63 1A 85 94 6B A3 A7 ].@].?\.IC.…"k£§
673. 00002A00:9E 9F A1 A3 9E A1 A5 A3 A4 A6 B9 B8 B7 EA E7 E5 žŸ¡£ž¡¥£¤¦¹¸•êçå
674. 00002A10:DF D6 CD CE C8 C2 B6 B0 A3 A8 9F 81 A7 9F 81 A2 ßÖÍÎÈÂ¶°£¨Ÿ□§Ÿ□¢
675. 00002A20:98 77 9A 8F 66 9A 91 6B A2 9B 84 B5 B4 AB D4 D1 ˜wš□Fš'k¢›„µ´«ÔÑ
676. 00002A30:CD DE D9 D3 DE D8 D1 D0 C9 C1 D0 CA C4 DF DD DC ÍÞÙÓÞØÑÐÉÁÐÊÄßÝÜ
677. 00002A40:E4 E1 DE CB C8 C6 7E 7A 71 57 52 49 56 53 50 5B äáÞËÈÆ~zqWRIVSP[
678. 00002A50:5C 5E 6E 71 74 86 88 8B A0 A1 A3 A6 AB A0 82 8F \^nqt†ˆ‹ ¡£¦« ‚□
679. 00002A60:66 4C 63 18 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B FLC.B[.B[.B[.B[.
680. 00002A70:42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 B[.B[.B[.B[.B[.B
681. 00002A80:5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 44 5A [.B[.B[.B[.B[.DZ
682. 00002A90:0B 44 5A 0B 42 5B 0B 42 5B 0B 42 5B 0B 43 5C 0C .DZ.B[.B[.B[.C\.
683. 00002AA0:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5B 09 43 C\.C\.C\.C\.C[.C
684. 00002AB0:5B 09 44 5A 09 44 5A 0B 43 59 0A 43 59 0A 43 59 [.DZ.DZ.CY.CY.CY
685. 00002AC0:0A 43 59 0A 43 5A 0A 00 40 5D 0D 40 5E 0E 41 5E .CY.CZ..@].@^.A^
686. 00002AD0:0E 41 5E 0E 41 5E 0E 41 5E 0E 40 5D 0D 40 5D 0D .A^.A^.A^.@].@].
687. 00002AE0:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 3F @].@].@].@].@].?
688. 00002AF0:5C 0C 3F 5C 0C 3F 5C 0C 3F 5C 0C 40 5D 0D 40 5D \.?\.?\.?\.@].@]
689. 00002B00:0D 40 5D 0D 3F 5C 0C 3F 5C 0C 44 60 13 58 70 2E .@].?\.?\.D`.Xp.
690. 00002B10:73 85 53 84 92 6B 8F 99 7C A6 AC 9A B7 B8 AE B9 s…S„'k□™|¦¬š•¸®¹
691. 00002B20:B9 B8 BB BD BE BA BD BF B8 BB BD B5 B8 BB B0 B3 ¹¸»½¾º½¿¸»½µ¸»°³
692. 00002B30:B6 A9 AB AD 9D 9E 9F 8F 8F 90 86 86 86 87 86 86 ¶©«­□žŸ□□□†††‡††
693. 00002B40:B5 B1 AD DE D8 D1 BB B5 AF 6C 6C 6C 79 79 79 6F µ±­ÞØÑ»µ¯lllyyyo
694. 00002B50:6F 6F 71 72 71 7A 7C 76 7D 83 73 7C 86 68 7B 86 ooqrqz|v}ƒs|†h{†
695. 00002B60:60 6F 7E 4D 57 6B 2A 46 5E 11 43 5C 0C 43 5D 0A `o~MWk\*F^.C\.C].
696. 00002B70:42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 B[.B[.B[.B[.B[.B
697. 00002B80:5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B [.B[.B[.B[.B[.B[
698. 00002B90:0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5A 0B 42 5A 0B .B[.B[.B[.BZ.BZ.
699. 00002BA0:42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 43 5C 0C 43 B[.B[.B[.B[.C\.C
700. 00002BB0:5C 0A 43 5C 0A 43 5C 0A 44 5B 09 44 5A 0A 44 5A \.C\.C\.D[.DZ.DZ
701. 00002BC0:0A 44 5A 0B 44 5A 0B 43 59 0A 43 59 0A 43 59 0A .DZ.DZ.CY.CY.CY.
702. 00002BD0:44 5A 0B 00 3E 5E 0D 3F 5D 0D 3F 5D 0D 40 5E 0E DZ..>^.?].?].@^.
703. 00002BE0:41 5E 0E 40 5D 0D 40 5D 0D 3F 5C 0C 40 5D 0C 40 A^.@].@].?\.@].@
704. 00002BF0:5D 0C 40 5D 0C 40 5D 0C 3F 5C 0B 3F 5C 0B 3F 5C ].@].@].?\.?\.?\
705. 00002C00:0B 3F 5C 0B 40 5D 0D 40 5D 0D 40 5D 0D 3F 5C 0C .?\.@].@].@].?\.
706. 00002C10:3F 5C 0C 3F 5C 0C 40 5D 0D 40 5D 0D 40 5D 0D 40 ?\.?\.@].@].@].@
707. 00002C20:5D 0D 40 5D 0D 41 5C 0C 48 62 16 52 69 23 58 6E ].@].A\.HB.Ri#Xn
708. 00002C30:2D 5D 72 34 61 75 3A 64 77 3F 67 79 42 68 7A 44 -]r4Au:Dw?gyBhzD
709. 00002C40:69 7B 46 69 7A 46 68 78 44 66 77 42 64 75 40 8C i{FizFhxDFwBDu@Œ
710. 00002C50:96 6D C0 BE B0 51 62 2B 57 6B 2D 58 6D 2C 51 67 –mÀ¾°QB+Wk-Xm,Qg
711. 00002C60:22 49 60 15 41 5B 0B 41 5B 0B 41 5B 0B 41 5B 0B "I`.A[.A[.A[.A[.
712. 00002C70:41 5B 0B 40 5B 0B 40 5B 0B 40 5C 09 42 5B 0A 42 A[.@[.@[.@\.B[.B
713. 00002C80:5B 0A 42 5B 0A 42 5B 0A 42 5B 0A 42 5B 0A 42 5B [.B[.B[.B[.B[.B[
714. 00002C90:0A 41 5A 0A 42 5A 0C 42 5A 0C 42 5A 0C 42 5B 0B .AZ.BZ.BZ.BZ.B[.
715. 00002CA0:42 5B 0B 42 5B 0B 42 5B 0A 42 5B 0A 44 5A 0B 44 B[.B[.B[.B[.DZ.D
716. 00002CB0:5A 0B 44 5A 0A 44 5A 0A 44 5A 0A 45 5B 0B 45 5B Z.DZ.DZ.DZ.E[.E[
717. 00002CC0:0B 45 5B 0B 44 5A 09 44 5A 09 44 5A 0B 44 5A 0B .E[.DZ.DZ.DZ.DZ.
718. 00002CD0:44 5A 0B 44 5A 0B 43 59 0A 43 59 0A 43 5B 09 00 DZ.DZ.CY.CY.C[..
719. 00002CE0:3F 5F 0D 3F 5F 0D 3F 5F 0D 3F 5F 0D 40 5E 0D 40 ?\_.?\_.?\_.?\_.@^.@
720. 00002CF0:5E 0D 40 5E 0D 40 5E 0D 40 5F 0C 40 5F 0C 40 5F ^.@^.@^.@\_.@\_.@\_
721. 00002D00:0C 40 5F 0C 3F 5E 0B 3F 5E 0B 40 5E 0B 40 5E 0B .@\_.?^.?^.@^.@^.
722. 00002D10:40 5D 0C 40 5D 0C 40 5D 0C 40 5D 0C 40 5D 0C 40 @].@].@].@].@].@
723. 00002D20:5D 0C 40 5D 0C 40 5D 0C 40 5D 0C 40 5D 0C 40 5D ].@].@].@].@].@]
724. 00002D30:0C 41 5D 0C 41 5C 0B 41 5C 0B 41 5C 0B 41 5C 0B .A].A\.A\.A\.A\.
725. 00002D40:41 5D 0C 41 5D 0C 41 5D 0C 40 5C 0B 40 5C 0B 40 A].A].A].@\.@\.@
726. 00002D50:5C 0B 40 5C 0B 40 5C 0C 41 5C 0C 41 5C 0C 5A 70 \.@\.@\.A\.A\.Zp
727. 00002D60:2B 41 5C 0C 41 5C 0C 41 5C 0C 41 5C 0C 41 5C 0C +A\.A\.A\.A\.A\.
728. 00002D70:41 5C 0B 41 5C 0B 41 5C 0B 41 5C 0B 41 5C 0B 41 A\.A\.A\.A\.A\.A
729. 00002D80:5C 0B 41 5C 0B 41 5D 0B 42 5D 0A 42 5D 0A 42 5D \.A\.A].B].B].B]
730. 00002D90:0A 42 5D 0A 42 5D 0A 42 5D 0A 42 5D 0A 42 5D 0A .B].B].B].B].B].
731. 00002DA0:42 5C 0B 42 5C 0B 42 5C 0B 42 5C 0B 42 5C 0B 43 B\.B\.B\.B\.B\.C
732. 00002DB0:5C 0B 43 5D 0A 43 5D 0A 44 5C 0B 44 5C 0B 44 5C \.C].C].D\.D\.D\
733. 00002DC0:0A 44 5C 0A 44 5C 0A 44 5C 0A 44 5C 0A 44 5C 0A .D\.D\.D\.D\.D\.
734. 00002DD0:43 5B 0A 43 5B 0A 43 5B 0A 43 5B 0A 43 5B 0A 43 C[.C[.C[.C[.C[.C
735. 00002DE0:5B 0A 43 5B 0A 43 5B 0A 43 5D 09 00 3F 61 0C 40 [.C[.C[.C]..?A.@
736. 00002DF0:62 0D 40 62 0D 3F 61 0C 3F 61 0C 3F 61 0C 3F 61 B.@B.?A.?A.?A.?A
737. 00002E00:0C 3F 61 0C 3F 61 0C 3F 61 0C 3F 61 0C 3F 61 0C .?A.?A.?A.?A.?A.
738. 00002E10:3F 60 0C 3F 60 0C 40 60 0C 40 60 0C 40 60 0C 40 ?`.?`.@`.@`.@`.@
739. 00002E20:60 0C 3F 5F 0C 3F 5F 0C 3F 5F 0C 40 60 0C 40 60 `.?\_.?\_.?\_.@`.@`
740. 00002E30:0C 40 60 0C 3F 5F 0B 3F 5F 0B 3F 5F 0C 41 5F 0C .@`.?\_.?\_.?\_.A\_.
741. 00002E40:41 5E 0B 41 5E 0B 42 5F 0C 42 5F 0C 40 60 0C 40 A^.A^.B\_.B\_.@`.@
742. 00002E50:60 0C 40 60 0C 3F 5F 0B 3F 5F 0B 3F 5F 0B 3F 5F `.@`.?\_.?\_.?\_.?\_
743. 00002E60:0B 3F 5F 0B 42 5F 0C 42 5F 0C 41 5E 0B 41 5E 0B .?\_.B\_.B\_.A^.A^.
744. 00002E70:41 5E 0B 41 5E 0B 41 5E 0B 41 5F 0C 41 5E 0B 41 A^.A^.A^.A\_.A^.A
745. 00002E80:5E 0B 41 5E 0B 41 5E 0B 41 5E 0B 41 5F 0B 41 5F ^.A^.A^.A^.A\_.A\_
746. 00002E90:0B 41 5F 0A 41 5F 0B 41 5F 0B 41 5F 0B 41 5F 0B .A\_.A\_.A\_.A\_.A\_.
747. 00002EA0:41 5F 0B 41 5F 0B 41 5F 0B 41 5F 0B 42 5F 0B 42 A\_.A\_.A\_.A\_.B\_.B
748. 00002EB0:5F 0B 41 5F 0B 41 5F 0B 41 5F 0B 43 5F 0B 43 5E \_.A\_.A\_.A\_.C\_.C^
749. 00002EC0:0A 43 5E 0A 43 5F 0B 43 5F 0B 43 5F 0B 43 5F 0B .C^.C\_.C\_.C\_.C\_.
750. 00002ED0:43 5F 0B 43 5F 0A 43 5F 0A 43 5F 0A 42 5E 09 42 C\_.C\_.C\_.C\_.B^.B
751. 00002EE0:5E 09 42 5E 09 42 5E 09 42 5E 09 42 5E 09 42 5E ^.B^.B^.B^.B^.B^
752. 00002EF0:09 42 5E 09 42 5E 09 00 3F 64 0B 3F 64 0B 3F 64 .B^.B^..?D.?D.?D
753. 00002F00:0B 3F 64 0B 3F 64 0B 3F 64 0B 3F 64 0B 3F 64 0B .?D.?D.?D.?D.?D.
754. 00002F10:3F 64 0C 3F 64 0C 3F 64 0C 3F 62 0C 40 62 0C 40 ?D.?D.?D.?B.@B.@
755. 00002F20:62 0C 40 62 0C 40 62 0C 40 62 0C 40 62 0C 3F 62 B.@B.@B.@B.@B.?B
756. 00002F30:0C 3F 62 0C 40 63 0C 40 63 0D 40 62 0C 40 62 0C .?B.@C.@C.@B.@B.
757. 00002F40:3F 62 0B 3F 62 0B 3F 62 0C 40 62 0C 40 62 0C 41 ?B.?B.?B.@B.@B.A
758. 00002F50:62 0C 41 63 0D 40 62 0C 40 63 0A 40 63 0A 40 63 B.AC.@B.@C.@C.@C
759. 00002F60:0A 3F 62 0C 3F 62 0C 3F 62 0C 3F 62 0C 3F 62 0C .?B.?B.?B.?B.?B.
760. 00002F70:40 61 0B 40 61 0B 40 61 0B 40 61 0B 40 61 0B 40 @A.@A.@A.@A.@A.@
761. 00002F80:61 0B 40 61 0B 41 62 0C 40 61 0B 40 61 0B 40 61 A.@A.AB.@A.@A.@A
762. 00002F90:0B 40 61 0B 40 61 0B 40 61 0B 40 61 0B 40 61 0B .@A.@A.@A.@A.@A.
763. 00002FA0:40 61 0B 40 61 0B 40 61 0B 40 61 0B 40 61 0B 40 @A.@A.@A.@A.@A.@
764. 00002FB0:61 0B 40 61 0B 40 61 0B 40 61 0B 40 61 0B 40 61 A.@A.@A.@A.@A.@A
765. 00002FC0:0B 40 62 09 40 62 09 40 62 09 42 61 09 42 61 09 .@B.@B.@B.BA.BA.
766. 00002FD0:42 61 09 42 61 09 42 61 09 42 61 09 42 61 09 42 BA.BA.BA.BA.BA.B
767. 00002FE0:61 09 42 61 09 42 61 09 40 62 09 42 61 09 42 61 A.BA.BA.@B.BA.BA
768. 00002FF0:09 42 61 09 42 61 09 42 61 09 42 61 09 42 61 09 .BA.BA.BA.BA.BA.
769. 00003000:42 61 09 00 3D 60 0F 3D 60 0F 3D 60 0F 3D 60 0F BA..=`.=`.=`.=`.
770. 00003010:3D 60 0F 3D 60 0F 3D 60 0F 3D 60 0F 3D 60 0F 3D =`.=`.=`.=`.=`.=
771. 00003020:60 0F 3D 60 0F 3D 60 0F 3D 60 0F 3D 60 0F 3D 60 `.=`.=`.=`.=`.=`
772. 00003030:0F 3D 60 0F 3D 61 0E 3D 61 0E 3C 60 0D 3C 60 0D .=`.=A.=A.<`.<`.
773. 00003040:3D 61 0E 3D 61 0E 3D 61 0E 3D 61 0E 3D 61 0E 3D =A.=A.=A.=A.=A.=
774. 00003050:61 0E 3D 61 0E 3D 61 0E 3D 61 0E 3D 61 0E 3D 61 A.=A.=A.=A.=A.=A
775. 00003060:0E 3D 61 0E 3D 61 0E 3D 61 0E 3D 61 0E 3D 61 0E .=A.=A.=A.=A.=A.
776. 00003070:3C 60 0D 3E 5F 0D 3E 5F 0D 3E 5F 0D 3E 5F 0D 3E <`.>\_.>\_.>\_.>\_.>
777. 00003080:5F 0D 3E 5F 0D 3E 5F 0D 3E 5F 0D 3E 5F 0D 3E 5F \_.>\_.>\_.>\_.>\_.>\_
778. 00003090:0D 3E 5F 0D 3E 5F 0D 3E 5F 0D 3E 5F 0D 3E 5F 0D .>\_.>\_.>\_.>\_.>\_.
779. 000030A0:3E 5F 0D 3E 5F 0D 3E 5F 0D 3E 5F 0D 3E 5F 0D 3E >\_.>\_.>\_.>\_.>\_.>
780. 000030B0:5F 0D 3E 5F 0D 3E 5F 0D 3F 5F 0D 3F 5F 0D 3F 5F \_.>\_.>\_.?\_.?\_.?\_
781. 000030C0:0D 3F 5F 0D 3E 5F 0D 3E 5F 0D 3E 5F 0D 3E 5F 0D .?\_.>\_.>\_.>\_.>\_.
782. 000030D0:3E 60 0B 3D 5F 0A 3E 5F 0A 3E 5F 0A 3F 60 0B 3F >`.=\_.>\_.>\_.?`.?
783. 000030E0:60 0B 3F 60 0B 3F 60 0B 3F 60 0B 3F 60 0B 3F 60 `.?`.?`.?`.?`.?`
784. 000030F0:0B 3F 60 0B 3E 60 0B 3E 60 0B 3E 60 0B 3F 60 0B .?`.>`.>`.>`.?`.
785. 00003100:3F 60 0B 3F 60 0B 3F 5F 0D 3F 5F 0D 3D 5F 0A 00 ?`.?`.?\_.?\_.=\_..
786. 00003110:12 00 00 00 0C 00 00 00 01 00 00 00 52 00 00 00 ............R...
787. 00003120:70 01 00 00 02 00 00 00 F3 FF FF FF 00 00 00 00 p.......óÿÿÿ....
788. 00003130:4E 0C 00 00 4E 0C 00 00 C8 00 00 00 00 00 00 01 N...N...È.......
789. 00003140:04 00 00 02 41 00 72 00 69 00 61 00 6C 00 00 00 ....A.r.i.A.l...
790. 00003150:00 00 00 00 00 00 00 00 00 00 00 00 0C 45 00 00 .............E..
791. 00003160:BC 16 E8 FE FE 07 00 00 20 00 CC 00 00 00 00 00 ¼.èþþ... .Ì.....
792. 00003170:4C 00 00 00 FE 07 00 00 DA 16 01 B8 FF FF FF FF L...þ...Ú..¸ÿÿÿÿ
793. 00003180:00 00 00 00 00 00 00 00 F0 F6 13 00 00 00 00 00 ........ðö......
794. 00003190:0E 20 05 27 00 00 00 00 28 00 00 00 00 00 00 00 . .'....(.......
795. 000031A0:1C 2F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ./..............
796. 000031B0:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
797. 000031C0:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
798. 000031D0:00 00 00 00 00 00 00 00 28 00 00 00 FF FF FF 00 ........(...ÿÿÿ.
799. 000031E0:1C 2F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ./..............
800. 000031F0:58 00 00 00 2C 00 00 00 00 00 00 00 00 00 00 00 X...,...........
801. 00003200:00 00 80 3F 00 00 00 00 00 00 00 00 00 00 80 3F ..€?..........€?
802. 00003210:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
803. 00003220:D0 C5 35 00 00 00 00 00 28 00 00 00 59 00 00 00 ÐÅ5.....(...Y...
804. 00003230:2D 00 00 00 01 00 18 00 00 00 00 00 1C 2F 00 00 -............/..
805. 00003240:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
806. 00003250:10 00 90 01 00 00 00 00 25 00 00 00 00 00 00 00 ..□.....%.......
807. 00003260:D3 3F EC FE FE 07 00 00 79 0D 21 11 00 00 00 00 Ó?ìþþ...y.!.....
808. 00003270:10 00 90 01 00 00 00 00 00 00 00 00 59 00 00 00 ..□.........Y...
809. 00003280:2D 00 00 00 64 76 00 08 00 00 00 00 25 00 00 00 -...Dv......%...
810. 00003290:0C 00 00 00 02 00 00 00 54 00 00 00 A0 00 00 00 ........T... ...
811. 000032A0:00 00 00 00 00 00 00 00 FF FF FF FF FF FF FF FF ........ÿÿÿÿÿÿÿÿ
812. 000032B0:01 00 00 00 AB 0A 0D 42 00 00 0D 42 12 00 00 00 ....«..B...B....
813. 000032C0:05 00 00 00 0E 00 00 00 4C 00 00 00 00 00 00 00 ........L.......
814. 000032D0:00 00 00 00 00 00 00 00 FF FF FF FF FF FF FF FF ........ÿÿÿÿÿÿÿÿ
815. 000032E0:68 00 00 00 53 00 69 00 6D 00 70 00 6C 00 65 00 h...S.i.m.p.l.E.
816. 000032F0:20 00 53 00 61 00 6D 00 70 00 6C 00 65 00 00 00 .S.A.m.p.l.E...
817. 00003300:09 00 00 00 03 00 00 00 0B 00 00 00 07 00 00 00 ................
818. 00003310:03 00 00 00 07 00 00 00 04 00 00 00 09 00 00 00 ................
819. 00003320:07 00 00 00 0B 00 00 00 07 00 00 00 03 00 00 00 ................
820. 00003330:07 00 00 00 09 00 00 00 52 00 00 00 70 01 00 00 ........R...p...
821. 00003340:03 00 00 00 F3 FF FF FF 00 00 00 00 4E 0C 00 00 ....óÿÿÿ....N...
822. 00003350:4E 0C 00 00 C8 00 00 00 00 00 00 00 04 00 00 02 N...È...........
823. 00003360:4D 00 69 00 63 00 72 00 6F 00 73 00 6F 00 66 00 M.i.C.r.o.s.o.F.
824. 00003370:74 00 20 00 53 00 61 00 6E 00 73 00 20 00 53 00 t. .S.A.n.s. .S.
825. 00003380:65 00 72 00 69 00 66 00 00 00 00 00 00 00 00 00 E.r.i.F.........
826. 00003390:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
827. 000033A0:00 00 00 00 28 00 00 00 59 00 00 00 2D 00 00 00 ....(...Y...-...
828. 000033B0:01 00 18 00 00 00 00 00 1C 2F 00 00 00 00 00 00 ........./......
829. 000033C0:00 00 00 00 00 00 00 00 00 00 00 00 10 00 90 01 ..............□.
830. 000033D0:00 00 00 00 25 00 00 00 00 00 00 00 D3 3F EC FE ....%.......Ó?ìþ
831. 000033E0:FE 07 00 00 79 0D 21 11 00 00 00 00 10 00 90 01 þ...y.!.......□.
832. 000033F0:00 00 00 00 00 00 00 00 59 00 00 00 2D 00 00 00 ........Y...-...
833. 00003400:64 76 00 08 00 00 00 00 00 00 00 00 00 08 0D 00 Dv..............
834. 00003410:0D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
835. 00003420:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
836. 00003430:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
837. 00003440:00 00 00 00 92 A0 CD 02 00 00 00 00 CA BE CD 02 ....' Í.....Ê¾Í.
838. 00003450:00 00 00 00 00 00 00 00 00 00 00 00 FF FF 5A FE ............ÿÿZþ
839. 00003460:00 00 00 00 00 00 00 00 00 00 00 00 95 F1 53 FE ............•ñSþ
840. 00003470:FE 07 00 00 BE 06 5A FE FE 07 00 00 87 F2 53 FE þ...¾.Zþþ...‡òSþ
841. 00003480:FE 07 00 00 C4 04 5A FE FE 07 00 00 79 0D 21 11 þ...Ä.Zþþ...y.!.
842. 00003490:00 00 00 00 01 00 00 00 00 00 00 00 F0 02 5A FE ............ð.Zþ
843. 000034A0:64 76 00 08 00 00 00 00 25 00 00 00 0C 00 00 00 Dv......%.......
844. 000034B0:03 00 00 00 52 00 00 00 70 01 00 00 04 00 00 00 ....R...p.......
845. 000034C0:F2 FF FF FF 00 00 00 00 4E 0C 00 00 4E 0C 00 00 òÿÿÿ....N...N...
846. 000034D0:C8 00 00 00 00 00 00 00 04 00 00 02 4D 00 69 00 È...........M.i.
847. 000034E0:63 00 72 00 6F 00 73 00 6F 00 66 00 74 00 20 00 C.r.o.s.o.F.t. .
848. 000034F0:53 00 61 00 6E 00 73 00 20 00 53 00 65 00 72 00 S.A.n.s. .S.E.r.
849. 00003500:69 00 66 00 00 00 00 00 00 00 00 00 00 00 00 00 i.F.............
850. 00003510:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
851. 00003520:C8 F1 13 00 00 00 00 00 10 00 90 01 00 00 00 00 Èñ........□.....
852. 00003530:20 AC CF 02 00 00 00 00 10 00 00 00 06 00 00 00 ¬Ï.............
853. 00003540:06 00 00 00 04 00 00 00 01 00 00 00 01 00 00 00 ................
854. 00003550:01 00 00 00 01 00 00 00 0D 00 00 00 00 00 00 00 ................
855. 00003560:03 00 00 00 00 08 00 00 3B 09 00 00 00 00 00 00 ........;.......
856. 00003570:A0 E8 07 02 00 00 00 00 03 00 00 00 FE 07 00 00 è..........þ...
857. 00003580:90 01 00 00 4D 00 69 00 63 00 72 00 6F 00 73 00 □...M.i.C.r.o.s.
858. 00003590:6F 00 66 00 74 00 20 00 53 00 61 00 00 00 73 00 o.F.t. .S.A...s.
859. 000035A0:20 00 53 00 65 00 72 00 69 00 66 00 00 00 00 00 .S.E.r.i.F.....
860. 000035B0:00 00 00 00 00 00 00 00 FF FF 5A FE 00 00 00 00 ........ÿÿZþ....
861. 000035C0:40 02 5A FE FE 07 00 00 9D 04 00 00 00 00 00 00 @.Zþþ...□.......
862. 000035D0:FF FF FF FF FF FF FF FF 01 00 00 00 00 00 00 00 ÿÿÿÿÿÿÿÿ........
863. 000035E0:20 AC CF 02 00 00 00 00 00 00 07 02 00 00 00 00 ¬Ï.............
864. 000035F0:10 AC CF 02 00 00 00 00 26 36 E3 76 00 00 00 00 .¬Ï.....&6ãv....
865. 00003600:00 00 07 02 00 00 00 00 01 00 00 00 00 27 00 00 .............'..
866. 00003610:00 00 00 00 00 00 00 00 20 AC CF 02 64 76 00 08 ........ ¬Ï.Dv..
867. 00003620:00 00 00 00 25 00 00 00 0C 00 00 00 04 00 00 00 ....%...........
868. 00003630:28 00 00 00 0C 00 00 00 03 00 00 00 52 00 00 00 (...........R...
869. 00003640:70 01 00 00 03 00 00 00 13 00 00 00 00 00 00 00 p...............
870. 00003650:4E 0C 00 00 4E 0C 00 00 C8 00 00 00 00 00 00 00 N...N...È.......
871. 00003660:04 00 00 02 4D 00 69 00 63 00 72 00 6F 00 73 00 ....M.i.C.r.o.s.
872. 00003670:6F 00 66 00 74 00 20 00 53 00 61 00 6E 00 73 00 o.F.t. .S.A.n.s.
873. 00003680:20 00 53 00 65 00 72 00 69 00 66 00 00 00 00 00 .S.E.r.i.F.....
874. 00003690:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
875. 000036A0:00 00 00 00 00 00 73 00 20 00 53 00 65 00 72 00 ......s. .S.E.r.
876. 000036B0:69 00 66 00 00 00 00 00 00 00 00 00 00 00 00 00 i.F.............
877. 000036C0:FF FF 5A FE 00 00 00 00 40 02 5A FE FE 07 00 00 ÿÿZþ....@.Zþþ...
878. 000036D0:9D 04 00 00 00 00 00 00 FF FF FF FF FF FF FF FF □.......ÿÿÿÿÿÿÿÿ
879. 000036E0:01 00 00 00 00 00 00 00 20 AC CF 02 00 00 00 00 ........ ¬Ï.....
880. 000036F0:00 00 07 02 00 00 00 00 40 02 5A FE FE 07 00 00 ........@.Zþþ...
881. 00003700:F3 14 00 00 00 00 00 00 F3 14 0A 1E 00 00 00 00 ó.......ó.......
882. 00003710:94 8A E8 FE FE 07 00 00 04 00 00 00 00 00 00 00 "Šèþþ...........
883. 00003720:65 58 53 FE 00 00 00 00 00 00 00 00 00 00 00 00 EXSþ............
884. 00003730:00 F5 13 00 00 00 00 00 03 01 56 E5 89 1A 00 00 .õ........Vå‰...
885. 00003740:55 00 00 00 00 00 00 00 00 00 00 00 04 00 00 00 U...............
886. 00003750:00 00 00 00 FE 07 00 00 79 0D 21 11 00 00 00 00 ....þ...y.!.....
887. 00003760:40 02 5A FE 00 00 00 00 26 06 5A FE FE 07 00 00 @.Zþ....&.Zþþ...
888. 00003770:08 F5 13 00 00 00 00 00 00 F5 13 00 00 00 00 00 .õ.......õ......
889. 00003780:07 CB 54 FE FE 07 00 00 79 0D 21 11 00 00 00 00 .ËTþþ...y.!.....
890. 00003790:04 00 00 00 00 00 00 00 24 07 5A FE FE 07 00 00 ........$.Zþþ...
891. 000037A0:01 00 00 00 64 76 00 08 00 00 00 00 25 00 00 00 ....Dv......%...
892. 000037B0:0C 00 00 00 03 00 00 00 25 00 00 00 0C 00 00 00 ........%.......
893. 000037C0:02 00 00 00 28 00 00 00 0C 00 00 00 04 00 00 00 ....(...........
894. 000037D0:28 00 00 00 0C 00 00 00 03 00 00 00 25 00 00 00 (...........%...
895. 000037E0:0C 00 00 00 0D 00 00 80 0E 00 00 00 14 00 00 00 .......€........
896. 000037F0:00 00 00 00 10 00 00 00 14 00 00 00 ............

### EMR\_HEADER Example

This section provides an example of an EMR\_HEADER record (section [2.3.4.2](#Section_de081cd7351f4cc2830bd03fb55e89ab)).

1. 00000000:01 00 00 00 D4 00 00 00 00 00 00 00 00 00 00 00
2. 00000010:59 00 00 00 59 00 00 00 00 00 00 00 00 00 00 00
3. 00000020:42 0C 00 00 41 0C 00 00 20 45 4D 46 00 00 01 00
4. 00000030:FC 37 00 00 16 00 00 00 05 00 00 00 34 00 00 00
5. 00000040:6C 00 00 00 00 00 00 00 80 07 00 00 B0 04 00 00
6. 00000050:A5 02 00 00 A7 01 00 00 00 00 00 00 00 00 00 00
7. 00000060:00 00 00 00 D5 55 0A 00 A5 75 06 00 53 00 61 00
8. 00000070:6D 00 70 00 6C 00 65 00 20 00 45 00 4D 00 46 00
9. 00000080:20 00 74 00 68 00 61 00 74 00 20 00 68 00 61 00
10. 00000090:73 00 20 00 61 00 20 00 62 00 72 00 75 00 73 00
11. 000000A0:68 00 20 00 66 00 69 00 6C 00 6C 00 2C 00 20 00
12. 000000B0:62 00 69 00 74 00 6D 00 61 00 70 00 2C 00 20 00
13. 000000C0:61 00 6E 00 64 00 20 00 74 00 65 00 78 00 74 00
14. 000000D0:00 00 00 00

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type (0x00000001) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size (0x000000D4) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... (0x00000059) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... (0x00000059) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Frame (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... (0x00000C42) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... (0x00000C31) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes)**: 0x00000001 identifies the record type as EMR\_HEADER.

**Size (4 bytes)**: 0x000000D4 is the record size in bytes.

**Bounds (16 bytes)**: 0x00000000, 0x00000000, 0x00000059, 0x00000059 specify the rectangular [**inclusive-inclusive**](#gt_c706d1bc-8878-4743-9b94-e9643c5269e4) bounds of the smallest rectangle that can be drawn around the image stored in the [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7) in logical units.

**Frame (16 bytes)**: 0x00000000, 0x00000000, 0x00000C42, 0x00000C31 specify the rectangular inclusive-inclusive dimensions, in .01 millimeter units, of a rectangle that surrounds the image stored in the metafile.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Signature (0x464D4520) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Version (0x00010000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Byte (0x000037FC) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Records (0x00000016) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Handles (0x0005) | | | | | | | | | | | | | | | | Reserved (0x0000) | | | | | | | | | | | | | | | |
| nDescription (0x00000034) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offDescription (0x0000006C) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PalEntries (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Signature (4 bytes)**: 0x464D4520 is the record signature, which consists of the [**ASCII**](#gt_79fa85ca-ac61-467c-b819-e97dc1a7a599) string "EMF".

**Version (4 bytes)**: 0x00010000 specifies [**EMF**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) metafile interoperability.

**Bytes (4 bytes)**: 0x000037FC specifies the size of the metafile in bytes.

**Records (4 bytes)**: 0x00000016 specifies the number of records in the metafile.

**Handles (2 bytes)**: 0x0005 specifies the number of graphics objects that are created during the processing of the metafile. These objects are referenced by their indexes in metafile records. Index values for created objects start at 1. This value can be used to compute the size needed for the EMF object table (section [3.1.1.1](#Section_e4fa4e6390964cdcb77685e2a1e4e1f4)).

**Reserved (2 bytes)**: 0x0000 is ignored.

**nDescription (4 bytes)**: 0x00000034 specifies the number of characters in the array that contains the description of the metafile's contents.

**offDescription (4 bytes)**: 0x0000006C specifies the offset from the beginning of this record to the array that contains the description of the metafile's contents.

**PalEntries (4 bytes)**: 0x00000000 specifies the number of entries in the metafile [**palette**](#gt_dca5cd91-8d01-424b-9722-c9e818562ee5). The location of the palette is specified in the EMR\_EOF record.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Device (0x00000780) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... (0x00000780) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Millimeters (0x000002A5) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... (0x000001A7) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbPixelFormat (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offPixelFormat (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| bOpenGL (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MicrometersX (0x000A55D5) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MicrometersY (0x000675A5) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EmfDescription ("Sample EMF that has a brush fill, bitmap, and text") | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Device (8 bytes)**: 0x00000780, 0x00000780 specify the size of the reference device, in pixels.

**Millimeters (8 bytes)**: 0x000002A5, 0x000001A7 specify the size of the reference device, in millimeters.

**cbPixelFormat (4 bytes)**: 0x00000000 specifies the size of the [PixelFormatDescriptor (section 2.2.22)](#Section_1db036d62da84b92b4f8e9cab8cc93b7) structure. This value indicates that no pixel format is defined.

**offPixelFormat (4 bytes)**: 0x00000000 specifies the offset to the PixelFormatDescriptor in the metafile. In this case, no pixel format structure is present.

**bOpenGL (4 bytes)**: 0x00000000 specifies that no [**OpenGL**](#gt_216becb9-a800-4a44-8206-2ec2231c9c67) commands are present in the metafile.

**Micrometers (8 bytes)**: 0x000A55D5, 0x000675A5 specify the horizontal and vertical size of the reference device, in micrometers.

**EmfDescription (4 bytes)**: "Sample EMF that has a brush fill, bitmap, and text".

### EMR\_CREATEBRUSHINDIRECT Example

This section provides an example of an [EMR\_CREATEBRUSHINDIRECT](#Section_b9a8ef5d00894e42b317e6ebc0ff098f) record (section 2.3.7.1).

1. 000000D0: 27 00 00 00 18 00 00 00 01 00 00 00
2. 000000E0:02 00 00 00 52 47 2A 00 03 00 00 00

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type (0x00000027) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size (0x00000018) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ihBrush (0x00000001) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| LogBrush (0x00000002) (12 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... (0x0052472A) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... (0x00000003) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes)**: 0x00000027 identifies this record type as EMR\_CREATEBRUSHINDIRECT.

**Size (4 bytes)**: 0x00000018 specifies the size of this record in bytes.

**ihBrush (4 bytes)**: 0x00000001 specifies the index of this brush object in the EMF object table (section [3.1.1.1)](#Section_e4fa4e6390964cdcb77685e2a1e4e1f4).

**LogBrush (12 bytes)**: A [LogBrushEx](#Section_6d790f8d51654746bac33443915c2071) object (section 2.2.12) that contains brush data.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| BrushStyle (0x00000002) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Color (0x0052472A) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BrushHatch (0x00000003) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**BrushStyle (4 bytes)**: 0x00000002 specifies a hatch brush style, from the **BrushStyle** enumeration ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.1.1.4).

**Color (4 bytes)**: 0x0052472A is a **ColorRef** object ([MS-WMF] section 2.2.2.8) that specifies the brush color value.

**BrushHatch (4 bytes)**: 0x00000003 specifies the brush hatch. Its interpretation depends on the value of **BrushStyle**. In this case, it specifies a 45-degree upward, left-to-right hatch pattern.

### EMR\_SELECTOBJECT Example 1

This section provides an example of an [EMR\_SELECTOBJECT](#Section_145b063d5f9641feb7ae1e615b2bc2bf) record (section 2.3.8.5).

1. 000000E0: 25 00 00 00
2. 000000F0:0C 00 00 00 01 00 00 00

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type (0x00000025) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size (0x0000000C) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ihObject (0x00000001) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes)**: 0x00000025 identifies this record type as EMR\_SELECTOBJECT.

**Size (4 bytes)**: 0x0000000C specifies the size of this record in bytes.

**ihObject (4 bytes)**: 0x00000001 specifies the index of an object in the EMF object table.

### EMR\_BITBLT Example 1

This section provides an example of an [EMR\_BITBLT record (section 2.3.1.2)](#Section_347d1c44184747ec87627059e9e9b185).

1. 000000F0: 4C 00 00 00 64 00 00 00
2. 00000100:00 00 00 00 00 00 00 00 59 00 00 00 59 00 00 00
3. 00000110:00 00 00 00 00 00 00 00 5A 00 00 00 5A 00 00 00
4. 00000120:21 00 F0 00 00 00 00 00 00 00 00 00 00 00 80 3F
5. 00000130:00 00 00 00 00 00 00 00 00 00 80 3F 00 00 00 00
6. 00000140:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
7. 00000150:00 00 00 00 00 00 00 00 00 00 00 00

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type (0x0000004C) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size (0x00000064) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... (0x00000059) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... (0x00000059) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| xDest (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| yDest (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cxDest (0x00000059) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cyDest (0x00000059) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes)**: 0x0000004C identifies this record type as EMR\_BITBLT.

**Size (4 bytes)**: 0x00000064 specifies the size of this record in bytes.

**Bounds (16 bytes)**: 0x00000000, 0x00000000, 0x00000059, 0x00000059 specify the bounding rectangle in logical units.

**xDest (4 bytes)**: 0x00000000 specifies the logical x-coordinate of the upper-left corner of the destination rectangle.

**yDest (4 bytes)**: 0x00000000 specifies the logical y-coordinate of the upper-left corner of the destination rectangle.

**cxDest (4 bytes)**: 0x0000005A specifies the logical width of the destination rectangle.

**cyDest (4 bytes)**: 0x0000005A specifies the logical height of the destination rectangle.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| BitBlitRasterOperation (0x00F00021) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| xSrc (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ySrc (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| xformSrc (0x3F800000) (24 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... (0x3F800000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... (0x00000059) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**BitBlitRasterOperation (4 bytes)**: 0x00F00021 specifies the [**raster operation**](#gt_772681cd-b9a5-440c-8021-b6219c70f6fc) code from the **Ternary Raster Operation** enumeration ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.1.1.31). This code defines how the color data of the source rectangle is to be combined with the color data of the destination rectangle to achieve the final color.

**xSrc (4 bytes)**: 0x00000000 specifies the logical x-coordinate of the upper-left corner of the source rectangle.

**ySrc (4 bytes)**: 0x00000000 specifies the logical y-coordinate of the upper-left corner of the source rectangle.

**xformSrc (24 bytes)**: 0x3F800000, 0x00000000, 0x00000000, 0x3F800000, 0x00000000, 0x00000000 specify the [**world-space**](#gt_baadb707-1343-413b-9b36-2496b06e9bd7) to [**page-space**](#gt_f530f123-858a-4af7-b626-adea0aeaa75a) [**transform**](#gt_fdca4a96-8e04-4dc0-acad-1d01e1384a5b). For more information on [**coordinate spaces**](#gt_84184077-d0d9-41e0-acaf-6596aa066813), see [[MSDN-WRLDPGSPC]](https://go.microsoft.com/fwlink/?LinkId=90169).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| BkColorSrc (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| UsageSrc (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offBmiSrc (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbBmiSrc (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offBitsSrc (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbBitsSrc (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**BkColorSrc (4 bytes)**: 0x00000000 specifies the background [**RGB**](#gt_2c716d3a-e60b-4e52-bbb0-2fdeb298003b) color.

**UsageSrc (4 bytes)**: 0x00000000 specifies the value of the **Colors** field of the **DeviceIndependentBitmap** object ([MS-WMF] section 2.2.2.9) from the [DIBColors](#Section_a5e722e3891a4a67be1aed5a48a7fda1) enumeration (section 2.1.9).

**offBmiSrc (4 bytes)**: 0x00000000 specifies the offset to the source **DeviceIndependentBitmap** object.

**cbBmiSrc (4 bytes)**: 0x00000000 specifies the size of the source **DeviceIndependentBitmap** object.

**offBitsSrc (4 bytes)**: 0x00000000 specifies the offset to the source [**bitmap**](#gt_26a17e58-eeba-4210-81b9-8bbadf7ee952) bits.

**cbBitsSrc (4 bytes)**: 0x00000000 specifies the size of the source bitmap bits.

### EMR\_SELECTOBJECT Example 2

This section provides an example of an [EMR\_SELECTOBJECT](#Section_145b063d5f9641feb7ae1e615b2bc2bf) record (section 2.3.8.5).

1. 00000150: 25 00 00 00
2. 00000160:0C 00 00 00 00 00 00 80

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type (0x00000025) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size (0x0000000C) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ihObject (0x80000000 = WHITE BRUSH) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes)**: 0x00000025 identifies this record type as EMR\_SELECTOBJECT.

**Size (4 bytes)**: 0x0000000C specifies the size of this record in bytes.

**ihObject (4 bytes)**: 0x80000000 specifies the index of an object in the EMF object table.

### EMR\_BITBLT Example 2

This section provides an example of an [EMR\_BITBLT record (section 2.3.1.2)](#Section_347d1c44184747ec87627059e9e9b185).

1. 00000160: 4C 00 00 00 A8 2F 00 00
2. 00000170:00 00 00 00 2D 00 00 00 59 00 00 00 59 00 00 00
3. 00000180:00 00 00 00 2D 00 00 00 5A 00 00 00 2D 00 00 00
4. 00000190:20 00 CC 00 00 00 00 00 00 00 00 00 00 00 80 3F
5. 000001A0:00 00 00 00 00 00 00 00 00 00 80 3F 00 00 00 00
6. 000001B0:00 00 00 00 FF FF FF 00 00 00 00 00 64 00 00 00
7. 000001C0:28 00 00 00 8C 00 00 00 1C 2F 00 00 28 00 00 00
8. 000001D0:59 00 00 00 2D 00 00 00 01 00 18 00 00 00 00 00
9. 000001E0:1C 2F 00 00 00 00 00 00 00 00 00 00 00 00 00 00
10. 000001F0:00 00 00 00 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E
11. 00000200:3F 5F 0E 3F 5F 0E 3F 5F 0E 3E 5E 0D 3F 5F 0E 3F
12. 00000210:5F 0E 41 5E 0E 41 5E 0E 41 5E 0E 41 5E 0E 41 5E
13. 00000220:0E 41 5E 0E 41 5E 0E 41 5E 0E 41 5E 0E 41 5E 0E
14. 00000230:41 5E 0E 41 5E 0E 41 5E 0E 41 5E 0E 40 5D 0D 3D
15. 00000240:59 0C 5A 60 4E AE AE AE BF BF BF C5 C5 C5 C0 C0
16. 00000250:C1 B1 B2 B1 A1 A1 A1 A9 AA AA AE AF B0 A2 A2 A2
17. 00000260:A6 A6 A4 AF AE AD AC AC AC A6 A6 A6 99 99 99 7D
18. 00000270:7D 7D 66 66 65 5A 5A 59 4F 4F 4F 58 58 58 76 76
19. 00000280:76 9E 9E 9E B5 B5 B5 3C 43 2D 32 46 0A 34 4A 0A
20. 00000290:34 49 0A 36 4C 0B 3A 52 0B 3F 59 0C 41 5C 0D 42
21. 000002A0:5D 0D 42 5D 0D 44 5D 0D 44 5D 0D 43 5C 0C 43 5C
22. 000002B0:0C 43 5C 0C 42 5B 0B 43 5C 0C 43 5C 0C 43 5C 0C
23. 000002C0:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43
24. 000002D0:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C
25. 000002E0:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C
26. 000002F0:45 5B 0C 45 5B 0C 45 5C 0A 45 5C 0A 45 5C 0A 00
27. 00000300:3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F
28. 00000310:5F 0E 3F 5F 0E 3E 5E 0D 3F 5F 0E 3F 5F 0E 41 5E
29. 00000320:0E 41 5E 0E 41 5E 0E 41 5E 0E 41 5E 0E 41 5E 0E
30. 00000330:41 5E 0E 41 5E 0E 41 5E 0E 41 5E 0E 41 5E 0E 41
31. 00000340:5E 0E 41 5E 0E 41 5E 0E 40 5D 0D 39 52 0C 70 73
32. 00000350:6B AD AD AD A8 A8 A8 99 99 98 A6 A6 A6 A6 A7 A7
33. 00000360:A3 A5 A6 A6 A5 A3 AE A5 9D C9 BB AD A1 9B 95 A2
34. 00000370:9F 9C B3 B1 AF B6 B4 B3 A9 A9 A9 97 93 90 8E 87
35. 00000380:81 89 84 7F 81 7E 7A 79 78 76 80 7F 7E 98 98 98
36. 00000390:BC BC BC 6D 70 65 33 47 0A 37 4D 0B 35 4B 0A 35
37. 000003A0:4B 0B 38 4F 0B 3D 56 0C 41 5B 0D 42 5D 0D 42 5D
38. 000003B0:0D 43 5D 0D 44 5D 0D 43 5C 0C 43 5C 0C 43 5C 0C
39. 000003C0:42 5B 0B 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43
40. 000003D0:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C
41. 000003E0:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C
42. 000003F0:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 45 5B 0C 45
43. 00000400:5B 0C 45 5C 0A 45 5C 0A 45 5C 0A 00 3F 5F 0E 3F
44. 00000410:5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F
45. 00000420:0E 3E 5E 0D 3F 5F 0E 3F 5F 0E 41 5E 0E 41 5E 0E
46. 00000430:41 5E 0E 41 5E 0E 41 5E 0E 41 5E 0E 41 5E 0E 41
47. 00000440:5E 0E 41 5E 0E 41 5E 0E 41 5E 0E 41 5E 0E 41 5E
48. 00000450:0E 41 5E 0E 40 5D 0D 38 51 0E 90 91 8E AC AC AC
49. 00000460:C3 C3 C3 C5 C6 C6 B9 BB BC B3 B0 AD BA AD 9E C8
50. 00000470:B4 A0 DB C8 B4 E4 D1 BE C2 B5 A9 82 7E 7A 8C 88
51. 00000480:84 83 80 7E 98 8F 85 BE AC 9A C9 B6 A3 CC BB A9
52. 00000490:CD C0 B3 AB A3 9B 80 7E 76 78 76 70 91 8F 8D 84
53. 000004A0:86 80 36 4B 0B 39 50 0B 38 4E 0B 37 4D 0B 38 4F
54. 000004B0:0B 3C 55 0C 40 5A 0D 42 5D 0D 42 5D 0D 42 5D 0D
55. 000004C0:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 42 5B 0B 43
56. 000004D0:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C
57. 000004E0:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C
58. 000004F0:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43
59. 00000500:5C 0C 43 5C 0C 43 5C 0C 45 5B 0A 45 5B 0A 45 5C
60. 00000510:0A 45 5C 0A 45 5C 0A 00 3F 5F 0E 3F 5F 0E 3F 5F
61. 00000520:0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3E 5E 0D
62. 00000530:3F 5F 0E 3F 5F 0E 41 5E 0E 41 5E 0E 41 5E 0E 41
63. 00000540:5E 0E 41 5E 0E 41 5E 0E 41 5E 0E 41 5E 0E 41 5E
64. 00000550:0E 41 5E 0E 41 5E 0E 41 5E 0E 41 5E 0E 41 5E 0E
65. 00000560:40 5D 0D 3D 50 1C 94 94 94 A1 A1 A0 C9 CA CA BC
66. 00000570:BB B8 BB AC 9C C4 AD 94 D2 BB A4 E0 CF BD E0 D0
67. 00000580:BF DA C7 B5 DE CB B8 9A 93 8B 7C 76 70 A2 93 83
68. 00000590:C0 AB 95 C6 B2 9D CA B8 A4 D1 C0 AD E0 D2 C4 E8
69. 000005A0:DC CF CD C3 B5 8A 85 74 72 72 61 83 82 7E 34 43
70. 000005B0:15 3C 54 0C 38 50 0B 38 4E 0B 39 50 0B 3D 55 0C
71. 000005C0:40 5B 0D 42 5D 0D 42 5D 0D 42 5D 0D 43 5C 0C 43
72. 000005D0:5C 0C 43 5C 0C 43 5C 0C 42 5B 0B 43 5C 0C 43 5C
73. 000005E0:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C
74. 000005F0:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43
75. 00000600:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C
76. 00000610:0C 43 5C 0A 45 5C 0A 45 5C 0A 45 5C 0A 45 5C 0A
77. 00000620:45 5C 0A 00 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E
78. 00000630:3F 5F 0E 3F 5F 0E 3F 5F 0E 3E 5E 0D 3F 5F 0E 3F
79. 00000640:5F 0E 41 5E 0E 41 5E 0E 41 5E 0E 41 5E 0E 41 5E
80. 00000650:0E 41 5E 0E 41 5E 0E 41 5E 0E 41 5E 0E 41 5E 0E
81. 00000660:41 5E 0E 41 5E 0E 41 5E 0E 41 5E 0E 40 5D 0D 4B
82. 00000670:56 33 B8 B8 B8 B7 B7 B7 A1 A1 A2 BC AC 9B CC B2
83. 00000680:97 D6 C3 AE E5 D6 C6 E4 D4 C4 E1 D2 C2 D9 C6 B4
84. 00000690:D7 C3 AD CE BF AF AD 9A 85 BB A5 8D BD A9 93 C3
85. 000006A0:AF 9B C8 B5 A2 CE BD AA DC CE C0 E2 D5 C9 E1 D4
86. 000006B0:C7 D5 C5 B4 B1 A9 95 91 97 7B 4A 57 2D 39 51 0B
87. 000006C0:38 50 0B 38 50 0B 3A 52 0C 3E 58 0C 41 5C 0D 42
88. 000006D0:5D 0D 42 5D 0D 42 5D 0D 41 5C 0C 43 5C 0C 43 5C
89. 000006E0:0C 43 5C 0C 42 5B 0B 43 5C 0C 43 5C 0C 43 5C 0C
90. 000006F0:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43
91. 00000700:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C
92. 00000710:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5D 0A
93. 00000720:45 5C 0A 45 5C 0A 45 5C 0A 45 5C 0A 45 5C 0A 00
94. 00000730:3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F
95. 00000740:5F 0E 3F 5F 0E 40 5D 0D 3F 5F 0E 3F 5F 0E 41 5E
96. 00000750:0F 41 5E 0F 41 5E 0F 41 5E 0E 41 5E 0E 40 5D 0D
97. 00000760:40 5D 0D 40 5D 0D 41 5E 0E 40 5D 0D 40 5D 0D 40
98. 00000770:5D 0D 40 5D 0D 41 5E 0E 41 5E 0E 61 6C 44 D0 D0
99. 00000780:D0 B0 B0 B0 C8 C3 BE DB C8 B3 DC CA B8 EB DD CF
100. 00000790:E8 D9 CA E5 D6 C7 E4 D6 C8 D9 C8 B5 D2 BF AA DD
101. 000007A0:CC BA C7 B3 9E B6 9F 89 BC A7 92 C1 AD 98 C6 B3
102. 000007B0:9F CC BA A8 DB CD BF DD D0 C2 DF D3 C6 E0 D3 C5
103. 000007C0:E2 D4 C4 A7 AA 8E 52 62 2B 37 4D 0B 39 51 0B 3B
104. 000007D0:53 0C 3E 57 0C 40 5A 0D 42 5D 0D 42 5D 0D 42 5D
105. 000007E0:0D 42 5D 0D 44 5D 0D 43 5C 0C 43 5C 0C 45 5B 0D
106. 000007F0:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43
107. 00000800:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C
108. 00000810:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C
109. 00000820:43 5C 0C 43 5C 0C 43 5C 0C 45 5C 0A 45 5C 0A 45
110. 00000830:5C 0A 45 5C 0A 45 5C 0A 45 5B 0C 00 3F 5F 0E 3F
111. 00000840:5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F
112. 00000850:0E 40 5D 0D 3F 5F 0E 3F 5F 0E 41 5E 0E 41 5E 0F
113. 00000860:41 5E 0E 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40
114. 00000870:5D 0D 41 5E 0E 40 5D 0D 40 5D 0D 40 5D 0D 40 5D
115. 00000880:0D 41 5E 0E 3F 5B 0D 6A 79 47 D1 D2 CE C5 C2 BF
116. 00000890:EF E6 DD E7 D8 CA EE E0 D3 ED DF D2 EA DC CE E9
117. 000008A0:DB CD E1 D1 C1 D6 C3 B0 D7 C4 B1 DA C9 B8 D5 C3
118. 000008B0:B1 B8 A3 8C BC A7 91 C0 AC 97 C5 B1 9D CB B9 A5
119. 000008C0:D6 C8 B8 CE BF AE D9 CB BD E1 D4 C7 E3 D4 C6 BC
120. 000008D0:B9 A2 53 63 2C 37 4C 0B 3C 55 0C 3E 58 0C 40 5B
121. 000008E0:0D 41 5C 0D 42 5D 0D 42 5D 0D 42 5D 0D 42 5D 0D
122. 000008F0:43 5C 0C 43 5C 0C 43 5C 0C 45 5B 0C 43 5C 0C 43
123. 00000900:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C
124. 00000910:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C
125. 00000920:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43
126. 00000930:5C 0C 43 5C 0C 45 5C 0A 45 5C 0A 45 5C 0A 45 5C
127. 00000940:0A 45 5C 0A 45 5B 0C 00 3F 5F 0E 3F 5F 0E 3F 5F
128. 00000950:0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 40 5D 0D
129. 00000960:3F 5F 0E 3F 5F 0E 41 5E 0E 41 5E 0E 40 5D 0D 40
130. 00000970:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 41 5E
131. 00000980:0E 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 41 5E 0E
132. 00000990:40 59 0F 75 84 54 A6 B1 8E DC D9 CE F5 EE E8 E7
133. 000009A0:DA CE E5 D7 C8 EB DD D0 EC DF D2 E4 D5 C4 D8 C8
134. 000009B0:B5 D9 CA B8 D9 CB BA D9 C8 B6 D9 C8 B6 C4 B0 9C
135. 000009C0:BA A5 90 BF AA 95 C3 AF 9B C9 B6 A3 C9 B9 A7 C7
136. 000009D0:B7 A6 D1 C2 B3 DC CF C0 E0 D2 C3 CE C4 B3 5B 69
137. 000009E0:33 3C 52 0D 40 5A 0D 41 5C 0D 42 5D 0D 42 5D 0D
138. 000009F0:42 5D 0D 42 5D 0D 41 5C 0C 42 5D 0D 43 5C 0C 43
139. 00000A00:5C 0C 43 5C 0C 45 5B 0C 43 5C 0C 43 5C 0C 43 5C
140. 00000A10:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C
141. 00000A20:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43
142. 00000A30:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C
143. 00000A40:0C 45 5C 0A 45 5C 0A 45 5C 0A 45 5C 0A 45 5C 0A
144. 00000A50:45 5B 0C 00 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E
145. 00000A60:3F 5F 0E 3F 5F 0E 3F 5F 0E 40 5D 0D 3F 5F 0E 3F
146. 00000A70:5F 0E 41 5E 0E 41 5E 0E 40 5D 0D 40 5D 0D 40 5D
147. 00000A80:0D 40 5D 0D 40 5D 0D 40 5D 0D 41 5E 0E 40 5D 0D
148. 00000A90:40 5D 0D 40 5D 0D 40 5D 0D 41 5E 0E 44 5C 12 8B
149. 00000AA0:97 6A 93 A2 76 C8 CA B5 F5 EE E9 E8 DC CF E0 D0
150. 00000AB0:C1 DF D2 C4 E1 BB A1 D4 AC 8E D5 B2 97 D3 A4 84
151. 00000AC0:D3 A5 86 D7 C4 B0 D7 C5 B1 CF BB A6 C2 AE 99 C1
152. 00000AD0:AD 98 C1 AE 99 C5 B0 9B C7 B5 A2 C8 B8 A7 D1 C2
153. 00000AE0:B2 E0 D3 C6 E4 D8 CA E2 D5 C6 71 7B 4B 42 58 10
154. 00000AF0:42 5D 0D 42 5D 0D 42 5D 0D 42 5D 0D 42 5D 0D 42
155. 00000B00:5D 0D 41 5C 0C 41 5C 0C 43 5C 0C 43 5C 0C 43 5C
156. 00000B10:0C 45 5B 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C
157. 00000B20:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43
158. 00000B30:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C
159. 00000B40:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 45 5C 0A
160. 00000B50:45 5C 0A 45 5C 0A 45 5C 0A 45 5C 0A 45 5B 0C 00
161. 00000B60:3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F
162. 00000B70:5F 0E 3F 5F 0E 40 5D 0D 3F 5F 0E 3F 5F 0E 41 5E
163. 00000B80:0E 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D
164. 00000B90:40 5D 0D 40 5D 0D 41 5E 0E 40 5D 0D 40 5D 0D 40
165. 00000BA0:5D 0D 40 5D 0D 41 5E 0E 4B 60 17 A0 AB 83 92 A2
166. 00000BB0:73 AC B6 94 F4 ED E7 E2 CF BE DA CA BA D7 C4 B2
167. 00000BC0:D0 86 58 CA 71 3D C9 69 32 C6 5D 24 C4 54 17 CC
168. 00000BD0:86 5D D5 C5 B2 D2 BC A6 C9 B3 9D C2 AD 96 C1 AC
169. 00000BE0:97 C8 B1 9A D9 C8 B6 D3 C4 B4 D1 C3 B3 DE D1 C4
170. 00000BF0:E0 D3 C5 E3 D5 C6 A3 A4 81 48 5E 13 42 5D 0D 42
171. 00000C00:5D 0D 42 5D 0D 42 5D 0D 42 5D 0D 42 5D 0D 41 5C
172. 00000C10:0C 41 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 45 5B 0C
173. 00000C20:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43
174. 00000C30:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C
175. 00000C40:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C
176. 00000C50:43 5C 0C 43 5C 0C 43 5C 0C 45 5C 0A 45 5C 0A 45
177. 00000C60:5C 0A 45 5C 0A 45 5C 0A 45 5B 0C 00 3F 5F 0E 3F
178. 00000C70:5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F
179. 00000C80:0E 40 5D 0D 3E 5E 0D 3E 5E 0D 40 5D 0D 40 5D 0D
180. 00000C90:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40
181. 00000CA0:5D 0D 41 5E 0E 40 5D 0D 40 5D 0D 40 5D 0D 40 5D
182. 00000CB0:0D 41 5E 0E 52 64 1F A4 AF 8A 95 A3 75 B3 AC 84
183. 00000CC0:DC AB 87 D7 A3 7D DB CD BD D2 A1 7F CD 76 42 CB
184. 00000CD0:71 3C C9 69 32 C7 62 2A C5 5B 20 C3 54 19 D8 AD
185. 00000CE0:90 E2 D7 CA D3 BE A9 CA B6 A0 D0 BD AA D1 BD A9
186. 00000CF0:DD CD BD DD D0 C2 D0 C1 B1 DA CD BE DE D1 C3 DB
187. 00000D00:CD BC 91 9A 6E 4E 62 18 42 5D 0D 42 5D 0D 42 5D
188. 00000D10:0D 42 5D 0D 42 5E 0B 41 5C 0C 41 5C 0C 41 5C 0C
189. 00000D20:43 5C 0C 43 5C 0C 43 5C 0C 45 5B 0C 43 5C 0C 43
190. 00000D30:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C
191. 00000D40:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C
192. 00000D50:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43
193. 00000D60:5C 0C 43 5C 0C 45 5C 0A 45 5C 0A 45 5C 0A 45 5C
194. 00000D70:0A 45 5C 0A 45 5B 0C 00 3F 5F 0E 3F 5F 0E 3F 5F
195. 00000D80:0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 40 5D 0D
196. 00000D90:3E 5D 0D 3E 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40
197. 00000DA0:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D
198. 00000DB0:0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0E
199. 00000DC0:58 67 26 A4 B1 8B B8 B1 8A D6 9F 74 D5 98 6B D7
200. 00000DD0:A4 7F DD CF BF D2 90 66 CD 77 43 CA 6E 38 C8 65
201. 00000DE0:2C C6 60 27 C6 5B 21 C3 56 1B CB 71 40 DD C8 B6
202. 00000DF0:D5 C3 AF E1 D2 C2 DF D0 C1 D9 C8 B7 E1 D3 C4 E4
203. 00000E00:D8 CA D1 C2 B2 E0 D4 C7 E7 DA CD AD AA 90 6F 7F
204. 00000E10:47 52 63 1F 41 5B 0D 41 5C 0C 41 5C 0C 42 5D 0D
205. 00000E20:42 5D 0B 41 5C 0C 41 5C 0C 41 5B 0C 42 5B 0C 43
206. 00000E30:5B 0C 43 5C 0C 43 5B 0B 43 5C 0C 43 5C 0C 43 5B
207. 00000E40:0C 43 5B 0C 43 5B 0C 43 5C 0C 43 5C 0C 43 5C 0C
208. 00000E50:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43
209. 00000E60:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C
210. 00000E70:0C 44 5C 0A 44 5C 0A 45 5B 0A 45 5B 0A 45 5B 0A
211. 00000E80:45 5B 0B 00 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E
212. 00000E90:3F 5F 0E 3F 5F 0E 3F 5F 0E 40 5D 0D 40 5D 0D 40
213. 00000EA0:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D
214. 00000EB0:0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 41 5E 0E
215. 00000EC0:40 5D 0D 40 5D 0D 40 5D 0D 42 5C 0E 60 6E 33 A9
216. 00000ED0:B6 91 CD B0 8B D7 A2 78 D6 99 6D D9 A8 86 E2 D7
217. 00000EE0:CA DE BC A2 CC 78 44 D0 85 57 D2 8B 61 CC 74 42
218. 00000EF0:CC 76 46 CD 7D 4F CC 76 47 CF 9C 7A D8 C9 B7 DF
219. 00000F00:D0 C0 DF CF BF DD CE BE E4 D7 C9 EB E0 D4 D1 C3
220. 00000F10:B3 DC CF C1 F1 E6 DA C8 C0 AE 72 80 4F 56 66 24
221. 00000F20:42 5B 0E 41 5C 0C 41 5C 0C 42 5D 0D 43 5C 0C 43
222. 00000F30:5C 0C 43 5C 0C 43 5B 0D 42 5A 0C 43 5B 0D 43 5C
223. 00000F40:0C 42 5B 0B 43 5C 0C 43 5C 0C 43 5B 0D 43 5B 0D
224. 00000F50:43 5B 0D 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43
225. 00000F60:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C
226. 00000F70:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C
227. 00000F80:43 5C 0C 45 5B 0C 45 5B 0C 45 5B 0C 45 5C 0A 00
228. 00000F90:3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F
229. 00000FA0:5F 0E 3F 5F 0E 40 5D 0D 40 5D 0D 40 5D 0D 40 5D
230. 00000FB0:0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D
231. 00000FC0:40 5D 0D 40 5D 0D 40 5D 0D 41 5E 0E 40 5D 0D 40
232. 00000FD0:5D 0D 40 5D 0D 43 5C 10 78 82 4C B9 BF 9C D8 AB
233. 00000FE0:84 D6 A3 7A D6 A2 7C DE BE A5 E7 DA CD E8 DA CB
234. 00000FF0:DA AA 89 E7 D9 CC DB AE 91 C7 63 2B C6 5D 24 C4
235. 00001000:56 1B C2 51 15 D1 A2 83 E1 D4 C5 E1 D2 C2 E1 D1
236. 00001010:C2 D1 C0 AD D1 C2 B1 E2 D6 C9 D3 C5 B5 D4 C6 B7
237. 00001020:E2 D6 C8 EA DE D2 95 9A 76 5B 6B 2A 44 5C 0F 41
238. 00001030:5C 0C 41 5C 0C 41 5C 0C 43 5C 0C 43 5C 0C 43 5C
239. 00001040:0C 43 5B 0D 42 5A 0C 43 5B 0D 43 5C 0C 42 5B 0B
240. 00001050:43 5C 0C 43 5C 0C 43 5B 0D 43 5B 0D 43 5B 0D 43
241. 00001060:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C
242. 00001070:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C
243. 00001080:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 45
244. 00001090:5B 0C 45 5B 0C 45 5B 0C 45 5C 0A 00 3F 5F 0E 3F
245. 000010A0:5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F
246. 000010B0:0E 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D
247. 000010C0:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40
248. 000010D0:5D 0D 40 5D 0D 41 5E 0E 40 5D 0D 40 5D 0D 40 5D
249. 000010E0:0D 47 5E 13 8E 97 67 C4 BF 9F CE A9 81 CA B7 99
250. 000010F0:E4 DA CF DB CC BC E9 DA CC E8 DA CC E9 DB CE E5
251. 00001100:CF BE CB 73 40 C6 5F 26 C6 5C 22 C3 52 15 CA 75
252. 00001110:45 D9 CA B9 E6 D8 CA E3 D5 C6 D1 BF AE C3 AF 9A
253. 00001120:CE BE AD D1 C3 B3 CC BD AD D2 C4 B5 D9 CC BD DD
254. 00001130:CE BF CE C6 B3 6A 78 3A 47 5C 11 41 5C 0C 41 5C
255. 00001140:0C 42 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5B 0D
256. 00001150:42 5A 0C 43 5B 0D 43 5C 0C 42 5B 0B 43 5C 0C 43
257. 00001160:5C 0C 43 5B 0D 43 5B 0D 43 5B 0D 43 5C 0C 43 5C
258. 00001170:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C
259. 00001180:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43
260. 00001190:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 45 5B 0C 45 5B
261. 000011A0:0C 45 5B 0C 45 5C 0A 00 3F 5F 0E 3F 5F 0E 3F 5F
262. 000011B0:0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 40 5D 0D
263. 000011C0:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40
264. 000011D0:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D
265. 000011E0:0D 41 5E 0E 40 5D 0D 40 5D 0D 40 5D 0D 4C 61 17
266. 000011F0:9C A5 7A D1 BF 9E D6 A5 7D E2 BD 9F E3 CF BE DB
267. 00001200:CB BB E9 DA CC E8 D9 CB EA DF D3 DC B0 93 C7 63
268. 00001210:29 C5 5C 22 C5 59 1F C5 5B 21 DD BC A4 DE D2 C4
269. 00001220:E6 D8 CA E8 DA CC CD BB A9 C6 B3 9F CE BD AB CA
270. 00001230:B8 A5 CF BE AC D5 C4 B2 BD B4 9D AB AC 8B B6 BA
271. 00001240:9A 78 86 4D 4A 5E 14 43 5C 0C 43 5C 0C 43 5C 0C
272. 00001250:43 5C 0C 43 5C 0C 43 5C 0C 43 5B 0D 42 5A 0C 43
273. 00001260:5B 0D 43 5C 0C 42 5B 0B 43 5C 0C 43 5C 0C 43 5B
274. 00001270:0D 42 5A 0C 43 5B 0D 43 5C 0C 43 5C 0C 43 5C 0C
275. 00001280:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43
276. 00001290:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C
277. 000012A0:0C 43 5C 0C 43 5C 0C 45 5B 0C 45 5B 0C 45 5B 0C
278. 000012B0:45 5C 0A 00 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E
279. 000012C0:3F 5F 0E 3F 5F 0E 3F 5F 0E 40 5D 0D 40 5D 0D 40
280. 000012D0:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D
281. 000012E0:0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 41 5E 0E
282. 000012F0:40 5D 0D 40 5D 0D 40 5D 0D 53 64 1D A2 AA 82 D7
283. 00001300:BC 99 D8 A7 7F DA A8 81 E3 CE BB E1 D2 C3 E8 D9
284. 00001310:CC E8 D9 CB E4 D6 C7 E6 D7 C9 E2 BC A3 D2 84 57
285. 00001320:C5 56 1A D4 9A 76 E6 DE D2 E0 D0 C1 E3 D5 C6 F0
286. 00001330:E4 D7 E1 D2 C3 CA B7 A4 D1 C1 B0 C9 B7 A3 D0 BD
287. 00001340:A9 D8 C6 B2 B5 AC 92 7B 89 53 75 89 4D 70 82 46
288. 00001350:53 61 1C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43
289. 00001360:5C 0C 43 5C 0C 43 5B 0D 42 5A 0C 43 5B 0D 43 5C
290. 00001370:0C 42 5B 0B 43 5C 0C 43 5C 0C 43 5B 0D 42 5A 0C
291. 00001380:42 5A 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43
292. 00001390:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C
293. 000013A0:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C
294. 000013B0:43 5C 0C 45 5B 0C 45 5B 0C 45 5B 0C 45 5C 0A 00
295. 000013C0:3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F
296. 000013D0:5F 0E 3F 5F 0E 40 5D 0D 40 5D 0D 40 5D 0D 40 5D
297. 000013E0:0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D
298. 000013F0:40 5D 0D 40 5D 0D 40 5D 0D 41 5E 0E 40 5D 0D 40
299. 00001400:5D 0D 41 5D 0E 59 67 24 A9 B2 8C D1 BB 98 D7 A5
300. 00001410:7C D4 B0 8D DE D4 C5 DA CA BB DE D1 C5 DB D0 C3
301. 00001420:DC CE BF E4 D5 C5 EC E2 D7 F0 E5 D9 E4 C4 AE D7
302. 00001430:C8 B9 D4 C7 BA D8 C9 BB E0 D2 C4 E9 DD D0 E3 D4
303. 00001440:C6 CF BD AB D2 C3 B3 C2 B2 A1 C6 B6 A3 CD BA A7
304. 00001450:CD BD A7 8B 97 65 7D 8E 55 74 85 49 5A 66 21 44
305. 00001460:5C 0E 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C
306. 00001470:0C 43 5B 0D 42 5A 0C 43 5B 0D 43 5C 0C 42 5B 0B
307. 00001480:43 5C 0C 43 5C 0C 43 5B 0D 42 5A 0C 42 5A 0C 42
308. 00001490:5B 0B 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C
309. 000014A0:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C
310. 000014B0:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 45
311. 000014C0:5B 0C 45 5B 0C 45 5B 0C 45 5C 0A 00 3F 5F 0E 3F
312. 000014D0:5F 0E 3F 5F 0C 3F 5F 0C 3F 5F 0C 3F 5F 0E 40 5E
313. 000014E0:0E 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D
314. 000014F0:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40
315. 00001500:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 43 5D
316. 00001510:0F 69 71 33 B3 BC 99 CD BA 96 D6 A4 79 A2 9B 72
317. 00001520:B3 AE 91 DE D0 C2 D9 C2 AF CE 9D 7C CF 8E 64 CE
318. 00001530:7D 4C D6 AB 8D DD D4 C9 DB D1 C6 DB CF C2 D8 CA
319. 00001540:BC CF C1 B2 D1 C4 B6 DB CF C3 D6 C8 BB C6 B6 A5
320. 00001550:CF C2 B4 D4 C8 BC A1 9B 81 69 6C 3C 60 63 2E 6C
321. 00001560:77 41 7A 88 54 74 83 4B 60 6C 27 45 5C 0E 43 5C
322. 00001570:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5B 0D
323. 00001580:42 5A 0C 43 5B 0D 43 5C 0C 42 5B 0B 43 5C 0C 43
324. 00001590:5C 0C 43 5B 0D 44 5A 0C 44 5A 0C 44 5B 0B 43 5C
325. 000015A0:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C
326. 000015B0:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43
327. 000015C0:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5B 0C 43 5B
328. 000015D0:0C 45 5B 0C 43 5C 0A 00 3F 5F 0E 3F 5F 0E 3F 60
329. 000015E0:0C 3F 60 0C 3F 60 0C 3F 5F 0E 41 5E 0E 40 5D 0D
330. 000015F0:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40
331. 00001600:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D
332. 00001610:0D 40 5D 0D 40 5D 0D 40 5D 0D 45 5E 10 7D 82 48
333. 00001620:BE C5 A4 B9 B0 8A BA B2 93 D1 C9 B6 DB AC 89 E6
334. 00001630:CC B7 E8 D2 C0 D1 85 55 CC 75 41 CA 6C 36 C6 6E
335. 00001640:3B B5 A4 8E D2 C6 B8 E7 D8 C9 E5 D6 C7 E0 D1 C2
336. 00001650:DA CC BD DB CD C0 D3 C6 B7 BE AF A0 D0 C3 B5 E9
337. 00001660:DE D2 EA E2 D6 C3 BE A9 8B 89 64 61 66 34 65 6D
338. 00001670:3E 68 76 45 63 6C 2D 47 5D 0F 43 5C 0C 43 5C 0C
339. 00001680:43 5C 0C 43 5C 0C 43 5C 0C 43 5B 0D 42 5A 0C 43
340. 00001690:5B 0D 43 5C 0C 42 5B 0B 43 5C 0C 43 5C 0C 43 5B
341. 000016A0:0D 45 5B 0D 45 5B 0C 45 5B 0C 43 5C 0C 42 5B 0B
342. 000016B0:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43
343. 000016C0:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C
344. 000016D0:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 45 5B 0C
345. 000016E0:43 5D 0A 00 3F 5F 0E 3F 5F 0E 3F 60 0C 3F 60 0C
346. 000016F0:3F 60 0C 3F 5F 0E 41 5E 0E 40 5D 0D 40 5D 0D 40
347. 00001700:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D
348. 00001710:0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D
349. 00001720:40 5D 0D 40 5D 0D 47 5E 12 91 93 61 BC C2 A1 C3
350. 00001730:C4 AB E8 E0 D4 DF B4 93 D4 96 68 DE B2 93 E0 BB
351. 00001740:A0 CF 81 50 CD 7A 48 CB 71 3C C8 63 2A D3 9A 75
352. 00001750:E1 D5 C7 DB CB BB D8 C7 B7 D7 C6 B5 D8 C7 B6 D8
353. 00001760:C7 B6 D3 C1 AF C5 B2 9E DB CD BE E9 DE D3 EF E5
354. 00001770:DB F6 EE E4 FA F0 E7 E3 D8 CA BB B0 9A A1 9D 83
355. 00001780:71 74 3B 4A 5E 12 43 5C 0C 43 5C 0C 43 5C 0C 43
356. 00001790:5C 0C 43 5C 0C 43 5B 0D 42 5A 0C 43 5B 0D 43 5C
357. 000017A0:0C 42 5B 0B 43 5B 0C 43 5B 0C 43 5C 0C 45 5B 0C
358. 000017B0:44 5A 0B 44 5A 0B 42 5B 0B 42 5B 0B 43 5C 0C 43
359. 000017C0:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C
360. 000017D0:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C
361. 000017E0:43 5C 0C 43 5C 0C 43 5C 0C 45 5B 0C 43 5C 0A 00
362. 000017F0:3F 5F 0E 3F 5F 0E 3F 60 0C 3F 60 0C 3F 60 0C 3F
363. 00001800:5F 0E 41 5E 0E 40 5D 0D 40 5D 0D 40 5D 0D 40 5D
364. 00001810:0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D
365. 00001820:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40
366. 00001830:5D 0D 4E 60 17 9B 9C 6F BB C2 9E D7 D5 C2 E0 C0
367. 00001840:A6 D7 A0 76 D6 9C 72 D8 A1 7A D4 95 6A D0 82 51
368. 00001850:CD 7A 47 CA 6D 38 D7 A7 88 DB BF AA D9 C9 B8 D3
369. 00001860:C3 B2 D0 BF AC D0 BE AC D1 C0 AE CF BD AA C6 B3
370. 00001870:9F D1 BF AD E2 D5 C7 E6 DC D0 EA DF D4 E9 DE D3
371. 00001880:E4 D8 CC DC CE C0 D1 C0 AF CC BD AA 73 79 3E 50
372. 00001890:5F 17 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C
373. 000018A0:0C 43 5B 0D 42 5A 0C 43 5B 0D 43 5C 0C 42 5B 0B
374. 000018B0:43 5B 0D 43 5B 0D 43 5C 0C 45 5B 0C 44 5A 0B 44
375. 000018C0:5A 0B 42 5B 0B 42 5B 0B 43 5C 0C 43 5C 0C 43 5C
376. 000018D0:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C
377. 000018E0:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43
378. 000018F0:5C 0C 43 5C 0C 45 5B 0C 45 5C 0A 00 3F 5F 0E 3F
379. 00001900:5F 0E 3F 60 0C 3F 60 0C 3F 60 0C 3F 5F 0E 41 5E
380. 00001910:0E 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D
381. 00001920:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40
382. 00001930:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 57 64
383. 00001940:1E A5 A5 7C B8 BE 9B D4 D1 BF E2 C5 AD D8 A2 78
384. 00001950:D7 9E 74 D8 A4 7E D2 8C 5E D0 83 53 CC 74 40 D1
385. 00001960:8D 62 DE D0 C2 DC CE C0 D4 C2 B1 CF BD AB D1 BF
386. 00001970:AD D4 C3 B2 D4 C3 B1 CF BE AB D1 BF AD DC CC BC
387. 00001980:DE D0 C3 DF D3 C6 DD D0 C3 DC CF C1 DF D3 C5 DA
388. 00001990:CC BF D3 C5 B6 90 8D 58 60 6D 27 56 64 1E 43 5C
389. 000019A0:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5B 0D
390. 000019B0:42 5A 0C 43 5B 0D 43 5C 0C 42 5B 0B 43 5B 0D 43
391. 000019C0:5B 0D 43 5C 0C 44 5A 0B 44 5A 0B 44 5A 0B 42 5B
392. 000019D0:0B 42 5B 0B 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C
393. 000019E0:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43
394. 000019F0:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C
395. 00001A00:0C 45 5B 0C 45 5C 0A 00 3F 5F 0E 3F 5F 0E 3F 60
396. 00001A10:0C 3F 60 0C 3F 60 0C 3F 5F 0E 41 5E 0E 40 5D 0D
397. 00001A20:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40
398. 00001A30:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D
399. 00001A40:0D 40 5D 0D 40 5D 0D 40 5D 0D 62 6B 26 B0 B1 8C
400. 00001A50:BB BE A0 E3 DD D0 E8 D7 C8 D8 A2 7A D7 A4 7D D5
401. 00001A60:9C 74 D1 87 57 CE 7C 4A D0 88 5C D7 C2 AF D4 C5
402. 00001A70:B5 D5 C4 B2 D9 C9 B8 DD CD BD DC CC BC DA C9 B9
403. 00001A80:D7 C6 B6 D7 C6 B5 DF D0 C0 D7 C7 B6 D7 C8 B9 D5
404. 00001A90:C7 B9 D9 CB BD E5 DA CD E4 D9 CC E0 D3 C6 D7 CD
405. 00001AA0:BD 7C 84 48 5B 6A 1F 5C 67 21 43 5C 0C 43 5C 0C
406. 00001AB0:43 5C 0C 43 5C 0C 43 5C 0C 43 5B 0D 42 5A 0C 43
407. 00001AC0:5B 0D 43 5C 0C 42 5B 0B 43 5B 0D 42 5A 0C 42 5B
408. 00001AD0:0B 44 5A 0B 44 5A 0B 44 5A 0B 42 5B 0B 42 5B 0B
409. 00001AE0:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43
410. 00001AF0:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C
411. 00001B00:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 45 5B 0C
412. 00001B10:45 5C 0A 00 3F 5F 0E 3F 5F 0E 3F 60 0C 3F 60 0C
413. 00001B20:3F 60 0C 3F 5F 0E 41 5E 0E 40 5D 0D 40 5D 0D 40
414. 00001B30:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D
415. 00001B40:0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D
416. 00001B50:40 5D 0D 41 5D 0D 6D 73 32 B9 BA 9B D6 D1 BF EC
417. 00001B60:E3 DA EE E7 DF E1 C0 A5 D7 AB 88 D7 A3 7E DA A7
418. 00001B70:84 DD B1 94 D9 C4 B1 D4 C5 B4 DB CB BB E9 DB CD
419. 00001B80:E9 DB CE E3 D5 C6 DE CF BF DC CD BC DF CF C0 E1
420. 00001B90:D1 C2 DF CF C0 DA CA B9 D6 C7 B8 D5 C7 B9 E8 DC
421. 00001BA0:D1 E8 DC D1 E4 D8 CC E5 D9 CC B2 AD 93 7D 89 53
422. 00001BB0:6A 77 32 60 6A 24 44 5C 0C 43 5C 0C 43 5C 0C 43
423. 00001BC0:5C 0C 43 5C 0C 43 5B 0D 42 5A 0C 43 5B 0D 43 5C
424. 00001BD0:0C 42 5B 0B 42 5A 0C 42 5A 0C 42 5B 0B 44 5A 0B
425. 00001BE0:44 5A 0B 44 5A 0B 42 5B 0B 42 5B 0B 43 5C 0C 43
426. 00001BF0:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C
427. 00001C00:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C
428. 00001C10:43 5C 0C 43 5C 0C 43 5C 0C 45 5B 0C 45 5C 0A 00
429. 00001C20:40 5D 0D 41 5E 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F
430. 00001C30:5F 0E 41 5E 0E 40 5D 0D 40 5D 0D 40 5D 0D 40 5D
431. 00001C40:0D 41 5E 0E 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D
432. 00001C50:41 5E 0E 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 41
433. 00001C60:5E 0F 77 79 3C CC CA B4 EB E2 D9 ED E4 DB EF E7
434. 00001C70:E0 EF E7 DE DE CC BB DD CD BC E8 DB CD E9 DC CF
435. 00001C80:DC CD BD E5 D6 C7 F1 E4 D8 F0 E3 D6 EA DC CE E5
436. 00001C90:D6 C7 E1 D2 C2 DF CF C0 DF CF C0 DD CD BD D9 C8
437. 00001CA0:B8 D7 C6 B5 E0 D3 C6 E6 DB CF EB E0 D5 E1 D5 C8
438. 00001CB0:DF D3 C7 E4 D8 CB A7 A3 88 77 82 4F 77 86 4B 6B
439. 00001CC0:71 30 44 5D 0E 41 5C 0C 43 5C 0C 43 5C 0C 43 5C
440. 00001CD0:0C 43 5C 0C 43 5C 0C 43 5B 0D 43 5B 0D 43 5B 0D
441. 00001CE0:42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 44 5A 0B 44
442. 00001CF0:5A 0B 42 5B 0B 42 5B 0B 44 5A 0B 43 5C 0C 43 5C
443. 00001D00:0C 43 5D 0A 43 5D 0A 43 5D 0A 43 5C 0C 43 5C 0C
444. 00001D10:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43
445. 00001D20:5C 0C 43 5C 0C 42 5B 0B 44 5B 09 00 40 5D 0D 41
446. 00001D30:5E 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 41 5E
447. 00001D40:0E 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 41 5E 0E
448. 00001D50:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40
449. 00001D60:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 47 5F 13 80 7F
450. 00001D70:46 DE DA C9 EF E6 DD EE E5 DD F0 E8 E0 F1 E8 E0
451. 00001D80:E2 D2 C3 DE CD BD E3 D3 C3 E3 D4 C5 E2 D3 C5 E6
452. 00001D90:D9 CA E5 D7 C9 E7 DA CB E9 DA CC E5 D6 C8 E2 D3
453. 00001DA0:C4 E0 D1 C2 DA CA B9 D3 C2 B1 CF BD AA CC B9 A6
454. 00001DB0:D6 C7 B8 E1 D5 C9 E7 DB CF DF D3 C6 E0 D3 C7 DE
455. 00001DC0:D1 C5 D4 C9 B9 7F 85 5B 75 83 4A 74 78 3A 4A 5F
456. 00001DD0:14 41 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C
457. 00001DE0:42 5B 0B 42 5B 0C 42 5B 0C 43 5B 0C 42 5B 0B 42
458. 00001DF0:5B 0B 42 5B 0B 42 5B 0B 43 5A 0B 43 5A 0B 42 5B
459. 00001E00:0B 42 5B 0B 44 5A 0B 43 5C 0C 43 5C 0C 43 5D 0A
460. 00001E10:43 5D 0A 43 5D 0A 43 5C 0C 43 5C 0C 43 5C 0C 43
461. 00001E20:5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C
462. 00001E30:0C 42 5B 0B 44 5B 09 00 40 5D 0D 41 5E 0E 3F 5F
463. 00001E40:0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 41 5E 0E 40 5D 0D
464. 00001E50:40 5D 0D 40 5D 0D 40 5D 0D 41 5E 0E 40 5D 0D 40
465. 00001E60:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D
466. 00001E70:0D 40 5D 0D 40 5D 0D 4B 61 15 8E 8B 57 CF CE B6
467. 00001E80:EA E2 D6 F0 E7 DF F1 E9 E1 F1 E8 E0 E4 D5 C5 E1
468. 00001E90:D1 C0 E0 D0 C0 E0 D0 C2 E4 D5 C6 E3 D4 C5 E3 D4
469. 00001EA0:C5 E5 D6 C8 E8 DB CD EB DD D0 EB DD CF E8 DA CC
470. 00001EB0:E0 D1 C2 D7 C6 B5 CE BB A9 C9 B5 A2 D3 C4 B5 D3
471. 00001EC0:C4 B6 DA CD BF E1 D5 C9 E3 D7 CB DD D1 C4 EA DF
472. 00001ED0:D4 B7 B4 9C 77 85 4D 76 7D 3E 4F 61 17 41 5C 0C
473. 00001EE0:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 42 5B 0B 42
474. 00001EF0:5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B
475. 00001F00:0B 42 5B 0B 42 5B 0B 42 5B 0B 43 5A 0B 43 5A 0B
476. 00001F10:44 5A 0B 43 5C 0C 43 5C 0C 43 5D 0A 43 5D 0A 43
477. 00001F20:5D 0A 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C
478. 00001F30:0C 43 5C 0C 43 5C 0C 43 5C 0C 42 5B 0B 43 5A 0B
479. 00001F40:44 5B 09 00 40 5D 0D 40 5D 0D 3F 5F 0E 3F 5F 0E
480. 00001F50:3F 5F 0E 3F 5F 0E 40 5D 0D 40 5D 0D 40 5D 0D 40
481. 00001F60:5D 0D 40 5D 0D 41 5E 0E 40 5D 0D 40 5D 0D 40 5D
482. 00001F70:0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D
483. 00001F80:40 5D 0D 50 63 1A A3 9F 71 CA C9 B1 E0 D9 CB F1
484. 00001F90:E8 E0 F2 EA E2 F2 E9 E0 E7 D8 C8 E4 D4 C4 E1 D0
485. 00001FA0:C1 E4 D5 C6 DD CD BD D7 C6 B6 DE CF BF E5 D6 C8
486. 00001FB0:EB DE D0 F0 E3 D6 F1 E5 D8 F0 E3 D6 ED DF D3 E9
487. 00001FC0:DB CD E8 DA CC E3 D4 C5 E3 D7 CA E4 D8 CC D9 CB
488. 00001FD0:BD D9 CB BD E0 D3 C7 DF D3 C6 E7 DC D0 E1 D8 C8
489. 00001FE0:80 8D 56 7A 82 45 55 64 1B 41 5C 0C 43 5C 0C 43
490. 00001FF0:5C 0C 43 5C 0C 42 5B 0B 42 5B 0B 42 5B 0B 42 5B
491. 00002000:0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B
492. 00002010:42 5B 0B 42 5B 0B 44 5A 0B 44 5A 0B 44 5A 0B 43
493. 00002020:5C 0C 43 5C 0C 43 5D 0A 43 5D 0A 43 5D 0A 43 5C
494. 00002030:0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C
495. 00002040:43 5C 0C 43 5C 0C 43 5A 0B 44 5A 0B 44 5B 09 00
496. 00002050:40 5D 0D 40 5D 0D 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F
497. 00002060:5F 0E 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D
498. 00002070:0D 41 5E 0E 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D
499. 00002080:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 58
500. 00002090:67 20 B2 AF 88 D8 D4 C5 F0 E7 DE F1 E9 E1 F3 EC
501. 000020A0:E4 F3 E9 E1 E9 DA CB E6 D6 C7 E4 D4 C4 E6 D8 C9
502. 000020B0:DF D0 C0 D5 C4 B3 DB CB BA E5 D7 C9 EB DD D0 EF
503. 000020C0:E2 D5 F1 E4 D8 F0 E4 D7 ED DF D2 EE E1 D4 E6 D8
504. 000020D0:C9 C1 AE 98 CA B9 A8 E8 DC D1 EC E1 D7 E7 DC D0
505. 000020E0:DD D0 C3 D4 C6 B7 D8 CB BC AE AC 8B 79 87 4D 7B
506. 000020F0:84 4A 5C 67 20 41 5C 0C 43 5C 0C 43 5C 0C 43 5C
507. 00002100:0C 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B
508. 00002110:42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42
509. 00002120:5B 0B 44 5A 0B 44 5A 0B 44 5A 0B 43 5C 0C 43 5C
510. 00002130:0C 43 5D 0A 43 5D 0A 43 5D 0A 43 5C 0C 43 5C 0C
511. 00002140:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 42
512. 00002150:5B 0B 44 5A 0B 44 5A 0B 44 5B 09 00 40 5D 0D 40
513. 00002160:5D 0D 3F 5F 0E 3F 5F 0E 3F 5F 0E 3F 5F 0E 40 5D
514. 00002170:0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 41 5E 0E
515. 00002180:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40
516. 00002190:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 64 6C 29 C3 BD
517. 000021A0:A0 F4 EC E3 F2 E9 E0 F3 EA E2 F5 EE E6 F3 EA E1
518. 000021B0:EB DC CD E8 D9 CA E5 D5 C6 E4 D4 C5 E3 D3 C3 D9
519. 000021C0:C8 B8 D3 C1 AF DC CC BC EB DD D0 EF E2 D6 F2 E5
520. 000021D0:DA F2 E5 D9 F4 E7 DB EA DC CE C7 B3 A0 B7 A2 8B
521. 000021E0:D1 C2 B3 EB E0 D6 E6 DA CF EA E0 D5 EF E4 DB E8
522. 000021F0:DC D1 D0 C5 B4 8A 8F 63 7C 8B 51 79 85 4A 64 69
523. 00002200:27 41 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 42 5B 0B
524. 00002210:42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42
525. 00002220:5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 44 5A
526. 00002230:0B 44 5A 0B 44 5A 0B 43 5C 0C 43 5C 0C 43 5D 0A
527. 00002240:43 5D 0A 43 5D 0A 43 5C 0C 43 5C 0C 43 5C 0C 43
528. 00002250:5C 0C 43 5C 0C 43 5C 0C 42 5B 0B 44 5A 0B 44 5A
529. 00002260:0B 44 5A 0B 44 5B 09 00 40 5D 0D 40 5D 0D 3F 5E
530. 00002270:0D 3F 5E 0D 3F 5E 0D 3F 5E 0D 40 5D 0D 40 5D 0D
531. 00002280:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40
532. 00002290:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D
533. 000022A0:0D 3F 5C 0C 40 5C 0D 72 74 35 CC C5 AC F4 EC E4
534. 000022B0:F1 E9 E0 F6 EE E6 F5 EE E6 F2 E9 DE ED DE CF EB
535. 000022C0:DC CE E3 D3 C3 D8 C7 B4 DB CB BA DF D0 BF D9 C8
536. 000022D0:B7 D9 C8 B7 ED DE D0 F4 E6 D9 F9 ED E0 FD F0 E4
537. 000022E0:FF F3 E7 E0 CF BD CA B6 A1 C7 B3 9D F3 E7 DC F4
538. 000022F0:EA E0 E7 DD D0 E7 DC D0 E8 DD D2 EF E5 DB F0 E7
539. 00002300:DD A5 A6 81 7C 8A 50 76 83 47 6F 71 31 43 5C 0E
540. 00002310:43 5C 0C 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42
541. 00002320:5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B
542. 00002330:0B 42 5B 0B 42 5B 0B 42 5B 0B 44 5A 0B 44 5A 0B
543. 00002340:43 5A 0B 42 5B 0B 42 5B 0B 43 5C 0A 43 5C 0A 43
544. 00002350:5C 0A 43 5C 0C 43 5C 0C 43 5C 0B 43 5C 0B 43 5C
545. 00002360:0C 43 5B 0C 42 5A 0A 43 59 0A 44 5A 0B 44 5A 0B
546. 00002370:43 5B 09 00 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D
547. 00002380:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40
548. 00002390:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D
549. 000023A0:0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 3F 5C 0C
550. 000023B0:43 5E 0F 80 7E 41 D3 CD BA F2 E9 E0 F4 EB E2 F4
551. 000023C0:EA E1 F0 E8 DF E5 D9 CD DD CE BE E4 D5 C7 DC CD
552. 000023D0:BF D9 CA BB D9 CA BB DA CC BD D2 C4 B6 CC BF B2
553. 000023E0:CC C1 B5 C8 BD B2 C3 B9 AF C0 B6 AB BA AE A2 AA
554. 000023F0:9A 8B A6 95 84 BB AD 9E D7 CE C5 D5 CA BF D3 C6
555. 00002400:B9 DC CF C1 E5 D7 C9 EB DF D2 E6 DC CF 8D 92 67
556. 00002410:7A 87 4D 73 82 45 79 79 39 49 5E 11 43 5C 0C 42
557. 00002420:5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B
558. 00002430:0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B
559. 00002440:42 5B 0B 42 5B 0B 44 5A 0B 44 5A 0B 42 5B 0B 42
560. 00002450:5B 0B 42 5B 0B 43 5C 0C 43 5C 0C 43 5C 0C 43 5C
561. 00002460:0C 43 5C 0C 43 5D 0A 43 5D 0A 43 5C 0C 44 5A 0B
562. 00002470:43 59 0A 43 59 0A 44 5A 0B 44 5A 0B 42 5C 09 00
563. 00002480:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40
564. 00002490:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D
565. 000024A0:0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D
566. 000024B0:40 5D 0D 40 5D 0D 40 5D 0D 3F 5C 0C 45 5F 11 90
567. 000024C0:8A 4F EA E1 D5 E8 DF D7 E7 DF D6 E7 E1 DA E5 E0
568. 000024D0:DB E6 E3 DF EA E8 E5 F0 EE ED F2 F2 F1 F1 F1 F1
569. 000024E0:F0 F0 F0 EB EB EB E3 E4 E4 D7 D7 D8 C2 C3 C4 A8
570. 000024F0:A9 AA 8B 8B 8C 6C 6C 6D 53 54 55 45 45 46 3E 3F
571. 00002500:3F 3B 3B 3B 3C 3B 3A 41 40 3F 4A 48 46 5B 57 54
572. 00002510:78 72 6C 9C 94 8B B4 AB A2 A2 9F 86 95 9C 6F 7A
573. 00002520:8A 4F 7B 7D 3D 4B 5F 13 43 5C 0C 42 5B 0B 42 5B
574. 00002530:0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B
575. 00002540:42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42
576. 00002550:5B 0B 44 5A 0B 44 5A 0B 42 5B 0B 42 5B 0B 42 5B
577. 00002560:0B 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C
578. 00002570:43 5D 0A 43 5D 0A 43 5C 0C 44 5A 0B 43 59 0A 43
579. 00002580:59 0A 44 5A 0B 44 5A 0B 42 5C 09 00 40 5D 0D 40
580. 00002590:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D
581. 000025A0:0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D
582. 000025B0:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40
583. 000025C0:5D 0D 40 5D 0D 3F 5C 0C 52 64 1D A1 92 80 C5 C5
584. 000025D0:C4 D2 D2 D2 E1 E2 E3 EB EC ED F2 F2 F3 F5 F5 F5
585. 000025E0:F7 F7 F6 F6 F6 F6 F5 F4 F3 F4 F3 F1 F1 EF EE F1
586. 000025F0:EF ED ED EC EA E7 E5 E3 DF DE DC D2 D1 D0 C4 C2
587. 00002600:C0 B7 B5 B2 AB A9 A6 A0 9D 9B 93 90 8D 89 88 87
588. 00002610:7E 7E 7F 72 72 71 6B 6A 6A 69 69 6A 69 6A 6B 6F
589. 00002620:71 73 83 86 89 A2 A3 A5 BB BA BA C4 C7 BF AE A4
590. 00002630:81 55 64 1C 43 5C 0C 42 5B 0B 42 5B 0B 42 5B 0B
591. 00002640:42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42
592. 00002650:5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 44 5A
593. 00002660:0B 44 5A 0B 42 5B 0B 42 5B 0B 42 5B 0B 43 5C 0C
594. 00002670:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5D 0A 43
595. 00002680:5D 0A 43 5C 0C 44 5A 0B 44 5A 0B 43 59 0A 43 59
596. 00002690:0A 44 5A 0B 42 5C 09 00 40 5D 0D 40 5D 0D 40 5D
597. 000026A0:0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D
598. 000026B0:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40
599. 000026C0:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D
600. 000026D0:0D 3F 5C 0C 75 82 5E 9E A0 A2 C6 C7 C8 DE DE DE
601. 000026E0:EA E9 E8 F2 F1 F0 F5 F3 F0 F5 F3 F0 F1 ED E9 EB
602. 000026F0:E6 E0 E9 E3 DD E6 E1 DA E0 DB D6 E9 E5 DD F6 F4
603. 00002700:F3 E9 E5 E1 E4 DF DA DC D7 D0 D2 CA C1 D4 CC C4
604. 00002710:DB D4 CC E2 DC D5 D4 CC C3 CF C7 BC F3 EF EB DF
605. 00002720:D9 D2 D9 D2 CB DC D7 D1 DB D8 D4 C9 C4 B8 B8 AF
606. 00002730:96 C3 BF B7 D0 D0 D0 D0 D1 D2 CF D1 D3 86 8F 6D
607. 00002740:43 5C 0C 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42
608. 00002750:5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B
609. 00002760:0B 42 5B 0B 42 5B 0B 42 5B 0B 44 5A 0B 44 5A 0B
610. 00002770:42 5B 0B 42 5B 0B 42 5B 0B 43 5C 0C 43 5C 0C 43
611. 00002780:5C 0C 43 5C 0C 43 5C 0C 43 5D 0A 43 5D 0A 44 5A
612. 00002790:0B 44 5A 0B 44 5A 0B 43 59 0A 43 59 0A 44 5A 0B
613. 000027A0:42 5C 09 00 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D
614. 000027B0:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40
615. 000027C0:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D
616. 000027D0:0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 3F 5C 0C
617. 000027E0:89 91 7B AD AC AC AD A4 84 A2 97 5E A7 9B 5F EE
618. 000027F0:E9 DA FC FC FC F8 F6 F4 EE E9 E4 E5 DE D7 E6 E1
619. 00002800:DB E4 E0 DB B8 AA 7D B7 AD 7C FF FF FF F2 F2 F2
620. 00002810:E5 E1 DD DE D9 D3 D6 CF C8 DB D6 D0 DB D6 D0 D9
621. 00002820:D3 CD D1 CA C2 BE B4 A8 BA AF A3 C6 BD B3 CE C6
622. 00002830:BE D1 CD C7 C5 B8 99 9A 8F 4D 91 88 43 9A 90 4E
623. 00002840:A5 99 61 B2 A9 8B CB CB CC 9D A4 90 43 5C 0C 42
624. 00002850:5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B
625. 00002860:0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B
626. 00002870:42 5B 0B 42 5B 0B 44 5A 0B 44 5A 0B 42 5B 0B 42
627. 00002880:5B 0B 42 5B 0B 43 5C 0C 43 5C 0C 43 5C 0C 43 5C
628. 00002890:0C 43 5C 0C 43 5D 0A 43 5D 0A 44 5A 0B 44 5A 0B
629. 000028A0:43 59 0A 43 59 0A 43 59 0A 44 5A 0B 42 5C 09 00
630. 000028B0:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40
631. 000028C0:5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D
632. 000028D0:0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D
633. 000028E0:40 5D 0D 40 5D 0D 40 5D 0D 3F 5C 0C 90 99 7F 7D
634. 000028F0:7C 7C 73 70 69 83 7D 64 93 88 5F A6 9C 78 EA E8
635. 00002900:E4 FB F9 F7 E7 E1 DA E1 D9 D1 D9 D0 BF AB 9F 68
636. 00002910:8D 84 3E 99 8F 4C C8 BB 90 E6 E0 CA F3 F0 EB E9
637. 00002920:E5 E1 E0 DA D4 E6 E2 DE DD D7 D2 CE C7 BF DF DA
638. 00002930:D5 E1 DC D7 D5 CE C6 D6 CF C7 CC C3 B4 BC B1 99
639. 00002940:96 8A 54 8C 80 47 90 84 56 8E 83 5D 89 82 6C 9A
640. 00002950:97 91 B2 B2 B3 92 9B 80 43 5C 0C 42 5B 0B 42 5B
641. 00002960:0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B
642. 00002970:42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42
643. 00002980:5B 0B 44 5A 0B 44 5A 0B 42 5B 0B 42 5B 0B 42 5B
644. 00002990:0B 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C
645. 000029A0:43 5D 0A 42 5C 09 44 5A 0B 44 5A 0B 43 59 0A 43
646. 000029B0:59 0A 44 5A 0B 44 5A 0B 42 5C 09 00 40 5D 0B 40
647. 000029C0:5D 0B 40 5E 0C 40 5E 0C 40 5E 0C 40 5E 0C 40 5E
648. 000029D0:0C 40 5D 0B 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D
649. 000029E0:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40
650. 000029F0:5D 0D 40 5D 0D 3F 5C 0C 49 63 1A 85 94 6B A3 A7
651. 00002A00:9E 9F A1 A3 9E A1 A5 A3 A4 A6 B9 B8 B7 EA E7 E5
652. 00002A10:DF D6 CD CE C8 C2 B6 B0 A3 A8 9F 81 A7 9F 81 A2
653. 00002A20:98 77 9A 8F 66 9A 91 6B A2 9B 84 B5 B4 AB D4 D1
654. 00002A30:CD DE D9 D3 DE D8 D1 D0 C9 C1 D0 CA C4 DF DD DC
655. 00002A40:E4 E1 DE CB C8 C6 7E 7A 71 57 52 49 56 53 50 5B
656. 00002A50:5C 5E 6E 71 74 86 88 8B A0 A1 A3 A6 AB A0 82 8F
657. 00002A60:66 4C 63 18 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B
658. 00002A70:42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42
659. 00002A80:5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 44 5A
660. 00002A90:0B 44 5A 0B 42 5B 0B 42 5B 0B 42 5B 0B 43 5C 0C
661. 00002AA0:43 5C 0C 43 5C 0C 43 5C 0C 43 5C 0C 43 5B 09 43
662. 00002AB0:5B 09 44 5A 09 44 5A 0B 43 59 0A 43 59 0A 43 59
663. 00002AC0:0A 43 59 0A 43 5A 0A 00 40 5D 0D 40 5E 0E 41 5E
664. 00002AD0:0E 41 5E 0E 41 5E 0E 41 5E 0E 40 5D 0D 40 5D 0D
665. 00002AE0:40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 40 5D 0D 3F
666. 00002AF0:5C 0C 3F 5C 0C 3F 5C 0C 3F 5C 0C 40 5D 0D 40 5D
667. 00002B00:0D 40 5D 0D 3F 5C 0C 3F 5C 0C 44 60 13 58 70 2E
668. 00002B10:73 85 53 84 92 6B 8F 99 7C A6 AC 9A B7 B8 AE B9
669. 00002B20:B9 B8 BB BD BE BA BD BF B8 BB BD B5 B8 BB B0 B3
670. 00002B30:B6 A9 AB AD 9D 9E 9F 8F 8F 90 86 86 86 87 86 86
671. 00002B40:B5 B1 AD DE D8 D1 BB B5 AF 6C 6C 6C 79 79 79 6F
672. 00002B50:6F 6F 71 72 71 7A 7C 76 7D 83 73 7C 86 68 7B 86
673. 00002B60:60 6F 7E 4D 57 6B 2A 46 5E 11 43 5C 0C 43 5D 0A
674. 00002B70:42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42
675. 00002B80:5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5B
676. 00002B90:0B 42 5B 0B 42 5B 0B 42 5B 0B 42 5A 0B 42 5A 0B
677. 00002BA0:42 5B 0B 42 5B 0B 42 5B 0B 42 5B 0B 43 5C 0C 43
678. 00002BB0:5C 0A 43 5C 0A 43 5C 0A 44 5B 09 44 5A 0A 44 5A
679. 00002BC0:0A 44 5A 0B 44 5A 0B 43 59 0A 43 59 0A 43 59 0A
680. 00002BD0:44 5A 0B 00 3E 5E 0D 3F 5D 0D 3F 5D 0D 40 5E 0E
681. 00002BE0:41 5E 0E 40 5D 0D 40 5D 0D 3F 5C 0C 40 5D 0C 40
682. 00002BF0:5D 0C 40 5D 0C 40 5D 0C 3F 5C 0B 3F 5C 0B 3F 5C
683. 00002C00:0B 3F 5C 0B 40 5D 0D 40 5D 0D 40 5D 0D 3F 5C 0C
684. 00002C10:3F 5C 0C 3F 5C 0C 40 5D 0D 40 5D 0D 40 5D 0D 40
685. 00002C20:5D 0D 40 5D 0D 41 5C 0C 48 62 16 52 69 23 58 6E
686. 00002C30:2D 5D 72 34 61 75 3A 64 77 3F 67 79 42 68 7A 44
687. 00002C40:69 7B 46 69 7A 46 68 78 44 66 77 42 64 75 40 8C
688. 00002C50:96 6D C0 BE B0 51 62 2B 57 6B 2D 58 6D 2C 51 67
689. 00002C60:22 49 60 15 41 5B 0B 41 5B 0B 41 5B 0B 41 5B 0B
690. 00002C70:41 5B 0B 40 5B 0B 40 5B 0B 40 5C 09 42 5B 0A 42
691. 00002C80:5B 0A 42 5B 0A 42 5B 0A 42 5B 0A 42 5B 0A 42 5B
692. 00002C90:0A 41 5A 0A 42 5A 0C 42 5A 0C 42 5A 0C 42 5B 0B
693. 00002CA0:42 5B 0B 42 5B 0B 42 5B 0A 42 5B 0A 44 5A 0B 44
694. 00002CB0:5A 0B 44 5A 0A 44 5A 0A 44 5A 0A 45 5B 0B 45 5B
695. 00002CC0:0B 45 5B 0B 44 5A 09 44 5A 09 44 5A 0B 44 5A 0B
696. 00002CD0:44 5A 0B 44 5A 0B 43 59 0A 43 59 0A 43 5B 09 00
697. 00002CE0:3F 5F 0D 3F 5F 0D 3F 5F 0D 3F 5F 0D 40 5E 0D 40
698. 00002CF0:5E 0D 40 5E 0D 40 5E 0D 40 5F 0C 40 5F 0C 40 5F
699. 00002D00:0C 40 5F 0C 3F 5E 0B 3F 5E 0B 40 5E 0B 40 5E 0B
700. 00002D10:40 5D 0C 40 5D 0C 40 5D 0C 40 5D 0C 40 5D 0C 40
701. 00002D20:5D 0C 40 5D 0C 40 5D 0C 40 5D 0C 40 5D 0C 40 5D
702. 00002D30:0C 41 5D 0C 41 5C 0B 41 5C 0B 41 5C 0B 41 5C 0B
703. 00002D40:41 5D 0C 41 5D 0C 41 5D 0C 40 5C 0B 40 5C 0B 40
704. 00002D50:5C 0B 40 5C 0B 40 5C 0C 41 5C 0C 41 5C 0C 5A 70
705. 00002D60:2B 41 5C 0C 41 5C 0C 41 5C 0C 41 5C 0C 41 5C 0C
706. 00002D70:41 5C 0B 41 5C 0B 41 5C 0B 41 5C 0B 41 5C 0B 41
707. 00002D80:5C 0B 41 5C 0B 41 5D 0B 42 5D 0A 42 5D 0A 42 5D
708. 00002D90:0A 42 5D 0A 42 5D 0A 42 5D 0A 42 5D 0A 42 5D 0A
709. 00002DA0:42 5C 0B 42 5C 0B 42 5C 0B 42 5C 0B 42 5C 0B 43
710. 00002DB0:5C 0B 43 5D 0A 43 5D 0A 44 5C 0B 44 5C 0B 44 5C
711. 00002DC0:0A 44 5C 0A 44 5C 0A 44 5C 0A 44 5C 0A 44 5C 0A
712. 00002DD0:43 5B 0A 43 5B 0A 43 5B 0A 43 5B 0A 43 5B 0A 43
713. 00002DE0:5B 0A 43 5B 0A 43 5B 0A 43 5D 09 00 3F 61 0C 40
714. 00002DF0:62 0D 40 62 0D 3F 61 0C 3F 61 0C 3F 61 0C 3F 61
715. 00002E00:0C 3F 61 0C 3F 61 0C 3F 61 0C 3F 61 0C 3F 61 0C
716. 00002E10:3F 60 0C 3F 60 0C 40 60 0C 40 60 0C 40 60 0C 40
717. 00002E20:60 0C 3F 5F 0C 3F 5F 0C 3F 5F 0C 40 60 0C 40 60
718. 00002E30:0C 40 60 0C 3F 5F 0B 3F 5F 0B 3F 5F 0C 41 5F 0C
719. 00002E40:41 5E 0B 41 5E 0B 42 5F 0C 42 5F 0C 40 60 0C 40
720. 00002E50:60 0C 40 60 0C 3F 5F 0B 3F 5F 0B 3F 5F 0B 3F 5F
721. 00002E60:0B 3F 5F 0B 42 5F 0C 42 5F 0C 41 5E 0B 41 5E 0B
722. 00002E70:41 5E 0B 41 5E 0B 41 5E 0B 41 5F 0C 41 5E 0B 41
723. 00002E80:5E 0B 41 5E 0B 41 5E 0B 41 5E 0B 41 5F 0B 41 5F
724. 00002E90:0B 41 5F 0A 41 5F 0B 41 5F 0B 41 5F 0B 41 5F 0B
725. 00002EA0:41 5F 0B 41 5F 0B 41 5F 0B 41 5F 0B 42 5F 0B 42
726. 00002EB0:5F 0B 41 5F 0B 41 5F 0B 41 5F 0B 43 5F 0B 43 5E
727. 00002EC0:0A 43 5E 0A 43 5F 0B 43 5F 0B 43 5F 0B 43 5F 0B
728. 00002ED0:43 5F 0B 43 5F 0A 43 5F 0A 43 5F 0A 42 5E 09 42
729. 00002EE0:5E 09 42 5E 09 42 5E 09 42 5E 09 42 5E 09 42 5E
730. 00002EF0:09 42 5E 09 42 5E 09 00 3F 64 0B 3F 64 0B 3F 64
731. 00002F00:0B 3F 64 0B 3F 64 0B 3F 64 0B 3F 64 0B 3F 64 0B
732. 00002F10:3F 64 0C 3F 64 0C 3F 64 0C 3F 62 0C 40 62 0C 40
733. 00002F20:62 0C 40 62 0C 40 62 0C 40 62 0C 40 62 0C 3F 62
734. 00002F30:0C 3F 62 0C 40 63 0C 40 63 0D 40 62 0C 40 62 0C
735. 00002F40:3F 62 0B 3F 62 0B 3F 62 0C 40 62 0C 40 62 0C 41
736. 00002F50:62 0C 41 63 0D 40 62 0C 40 63 0A 40 63 0A 40 63
737. 00002F60:0A 3F 62 0C 3F 62 0C 3F 62 0C 3F 62 0C 3F 62 0C
738. 00002F70:40 61 0B 40 61 0B 40 61 0B 40 61 0B 40 61 0B 40
739. 00002F80:61 0B 40 61 0B 41 62 0C 40 61 0B 40 61 0B 40 61
740. 00002F90:0B 40 61 0B 40 61 0B 40 61 0B 40 61 0B 40 61 0B
741. 00002FA0:40 61 0B 40 61 0B 40 61 0B 40 61 0B 40 61 0B 40
742. 00002FB0:61 0B 40 61 0B 40 61 0B 40 61 0B 40 61 0B 40 61
743. 00002FC0:0B 40 62 09 40 62 09 40 62 09 42 61 09 42 61 09
744. 00002FD0:42 61 09 42 61 09 42 61 09 42 61 09 42 61 09 42
745. 00002FE0:61 09 42 61 09 42 61 09 40 62 09 42 61 09 42 61
746. 00002FF0:09 42 61 09 42 61 09 42 61 09 42 61 09 42 61 09
747. 00003000:42 61 09 00 3D 60 0F 3D 60 0F 3D 60 0F 3D 60 0F
748. 00003010:3D 60 0F 3D 60 0F 3D 60 0F 3D 60 0F 3D 60 0F 3D
749. 00003020:60 0F 3D 60 0F 3D 60 0F 3D 60 0F 3D 60 0F 3D 60
750. 00003030:0F 3D 60 0F 3D 61 0E 3D 61 0E 3C 60 0D 3C 60 0D
751. 00003040:3D 61 0E 3D 61 0E 3D 61 0E 3D 61 0E 3D 61 0E 3D
752. 00003050:61 0E 3D 61 0E 3D 61 0E 3D 61 0E 3D 61 0E 3D 61
753. 00003060:0E 3D 61 0E 3D 61 0E 3D 61 0E 3D 61 0E 3D 61 0E
754. 00003070:3C 60 0D 3E 5F 0D 3E 5F 0D 3E 5F 0D 3E 5F 0D 3E
755. 00003080:5F 0D 3E 5F 0D 3E 5F 0D 3E 5F 0D 3E 5F 0D 3E 5F
756. 00003090:0D 3E 5F 0D 3E 5F 0D 3E 5F 0D 3E 5F 0D 3E 5F 0D
757. 000030A0:3E 5F 0D 3E 5F 0D 3E 5F 0D 3E 5F 0D 3E 5F 0D 3E
758. 000030B0:5F 0D 3E 5F 0D 3E 5F 0D 3F 5F 0D 3F 5F 0D 3F 5F
759. 000030C0:0D 3F 5F 0D 3E 5F 0D 3E 5F 0D 3E 5F 0D 3E 5F 0D
760. 000030D0:3E 60 0B 3D 5F 0A 3E 5F 0A 3E 5F 0A 3F 60 0B 3F
761. 000030E0:60 0B 3F 60 0B 3F 60 0B 3F 60 0B 3F 60 0B 3F 60
762. 000030F0:0B 3F 60 0B 3E 60 0B 3E 60 0B 3E 60 0B 3F 60 0B
763. 00003100:3F 60 0B 3F 60 0B 3F 5F 0D 3F 5F 0D 3D 5F 0A 00

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type (0x0000004C) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size (0x00002FA8) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... (0x0000002D) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... (0x00000059) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... (0x00000059) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| xDest (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| yDest (0x0000002D) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cxDest (0x0000005A) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cyDest (0x0000002D) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes)**: 0x0000004C identifies this record type as EMR\_BITBLT.

**Size (4 bytes)**: 0x00002FA8 specifies the size of this record in bytes.

**Bounds (16 bytes)**: 0x00000000, 0x00000002D, 0x00000059, 0x00000059 specifies the bounding rectangle in logical units.

**xDest (4 bytes)**: 0x00000000 specifies the logical x-coordinate of the upper-left corner of the destination rectangle.

**yDest (4 bytes)**: 0x0000002D specifies the logical y-coordinate of the upper-left corner of the destination rectangle.

**cxDest (4 bytes)**: 0x0000005A specifies the logical width of the destination rectangle.

**cyDest (4 bytes)**: 0x0000002D specifies the logical height of the destination rectangle.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| BitBlitRasterOperation (0x00CC0020) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| xSrc (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ySrc (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| xformSrc (0x3F800000) (24 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... (0x3F800000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**BitBlitRasterOperation (4 bytes)**: 0x00CC0020 specifies the [**raster operation**](#gt_772681cd-b9a5-440c-8021-b6219c70f6fc) code from the **Ternary Raster Operation** enumeration ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.1.1.31). This code defines how the color data of the source rectangle is to be combined with the color data of the destination rectangle to achieve the final color.

**xSrc (4 bytes)**: 0x00000000 specifies the logical x-coordinate of the upper-left corner of the source rectangle.

**ySrc (4 bytes)**: 0x00000000 specifies the logical y-coordinate of the upper-left corner of the source rectangle.

**xformSrc (24 bytes)**: 0x3F800000, 0x00000000, 0x00000000, 0x3F800000, 0x00000000, 0x00000000 specify the [**world-space**](#gt_baadb707-1343-413b-9b36-2496b06e9bd7) to [**page-space**](#gt_f530f123-858a-4af7-b626-adea0aeaa75a) [**transform**](#gt_fdca4a96-8e04-4dc0-acad-1d01e1384a5b). For more information about [**coordinate spaces**](#gt_84184077-d0d9-41e0-acaf-6596aa066813), see [[MSDN-WRLDPGSPC]](https://go.microsoft.com/fwlink/?LinkId=90169).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| BkColorSrc (0x00FFFFFF) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| UsageSrc (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offBmiSrc (0x00000064) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbBmiSrc (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offBitsSrc (0x0000008C) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbBitsSrc (0x00002F1C) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**BkColorSrc (4 bytes)**: 0x00FFFFFF specifies the background [**RGB**](#gt_2c716d3a-e60b-4e52-bbb0-2fdeb298003b) color.

**UsageSrc (4 bytes)**: 0x00000000 specifies the value of the **Colors** field of the **DeviceIndependentBitmap** object ([MS-WMF] section 2.2.2.9) from the [DIBColors](#Section_a5e722e3891a4a67be1aed5a48a7fda1) enumeration (section 2.1.9).

**offBmiSrc (4 bytes)**: 0x00000064 specifies the offset to the source **DeviceIndependentBitmap** object.

**cbBmiSrc (4 bytes)**: 0x00000000 specifies the size of the source **DeviceIndependentBitmap** object.

**offBitsSrc (4 bytes)**: 0x0000008C specifies the offset to the source [**bitmap**](#gt_26a17e58-eeba-4210-81b9-8bbadf7ee952) bits.

**cbBitsSrc (4 bytes)**: 0x00002F1C specifies the size of the source bitmap bits.

### EMR\_SETBKMODE Example

This section provides an example of an [EMR\_SETBKMODE](#Section_7c584a42b6dd47b8a0b6f529133292f4) record (section 2.3.11.11).

1. 00003110:12 00 00 00 0C 00 00 00 01 00 00 00

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type (0x00000012) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size (0x0000000C) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Mode (0x00000001) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes)**: 0x00000012 identifies this record type as EMR\_SETBKMODE.

**Size (4 bytes)**: 0x0000000C specifies the size of this record in bytes.

**Mode (4 bytes)**: 0x00000001 specifies the background color value.

### EMR\_EXTCREATEFONTINDIRECTW Example 1

This section provides an example of an EMR\_EXTCREATEFONTINDIRECTW record (section [2.3.7.8](#Section_7e266b6d32e54201b6878ec40c24cd73)).

1. 00003110: 52 00 00 00
2. 00003120:70 01 00 00 02 00 00 00 F3 FF FF FF 00 00 00 00
3. 00003130:4E 0C 00 00 4E 0C 00 00 C8 00 00 00 00 00 00 01
4. 00003140:04 00 00 02 41 00 72 00 69 00 61 00 6C 00 00 00
5. 00003150:00 00 00 00 00 00 00 00 00 00 00 00 0C 45 00 00
6. 00003160:BC 16 E8 FE FE 07 00 00 20 00 CC 00 00 00 00 00
7. 00003170:4C 00 00 00 FE 07 00 00 DA 16 01 B8 FF FF FF FF
8. 00003180:00 00 00 00 00 00 00 00 F0 F6 13 00 00 00 00 00
9. 00003190:0E 20 05 27 00 00 00 00 28 00 00 00 00 00 00 00
10. 000031A0:1C 2F 00 00 00 00 00 00 00 00 00 00 00 00 00 00
11. 000031B0:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
12. 000031C0:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
13. 000031D0:00 00 00 00 00 00 00 00 28 00 00 00 FF FF FF 00
14. 000031E0:1C 2F 00 00 00 00 00 00 00 00 00 00 00 00 00 00
15. 000031F0:58 00 00 00 2C 00 00 00 00 00 00 00 00 00 00 00
16. 00003200:00 00 80 3F 00 00 00 00 00 00 00 00 00 00 80 3F
17. 00003210:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
18. 00003220:D0 C5 35 00 00 00 00 00 28 00 00 00 59 00 00 00
19. 00003230:2D 00 00 00 01 00 18 00 00 00 00 00 1C 2F 00 00
20. 00003240:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
21. 00003250:10 00 90 01 00 00 00 00 25 00 00 00 00 00 00 00
22. 00003260:D3 3F EC FE FE 07 00 00 79 0D 21 11 00 00 00 00
23. 00003270:10 00 90 01 00 00 00 00 00 00 00 00 59 00 00 00
24. 00003280:2D 00 00 00 64 76 00 08 00 00 00 00

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type (0x00000052) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size (0x00000170) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ihFonts (0x00000002) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| elw (360 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes)**: 0x00000052 identifies this record type as EMR\_EXTCREATEFONTINDIRECTW.

**Size (4 bytes)**: 0x00000170 specifies the size of this record in bytes.

**ihFonts (4 bytes)**: 0x00000002 specifies the object index that is assigned to the font.

**elw (360 bytes)**: To determine the type of logical font object in this field, an algorithm (section 2.3.7.8) is applied, which indicates this is a [LogFontExDv object (section 2.2.15)](#Section_595fcaaebc3741489afe859e81ca7f76).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Height (0xFFFFFFF3) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Width (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Escapement (0x00000C4E) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Orientation (0x00000C4E) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Weight 0x000000C8) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Italic (0x00) | | | | | | | | Underline (0x00) | | | | | | | | StrikeOut (0x00) | | | | | | | | CharSet (0x01) | | | | | | | |
| OutPrecision (0x04) | | | | | | | | ClipPrecision (0x00) | | | | | | | | Quality (0x00) | | | | | | | | PitchAndFamily (0x02) | | | | | | | |
| Facename ("Arial") (68 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Height (4 bytes)**: 0xFFFFFFF3 has an absolute value of 13, which specifies the character height for this font, in logical units.

**Width (4 bytes)**: 0x00000000 specifies a computed font width. The aspect ratio of the device is matched against the digitization aspect ratio of the font to find the closest match, determined by the absolute value of the difference.

**Escapement (4 bytes)**: 0x00000C4E specifies an angle of 315 degrees between the [**baseline**](#gt_81ddcdfb-c66c-4c83-b963-18da9456fa62) of a row of text and the x-axis of the device.

**Orientation (4 bytes)**: 0x00000C4E specifies an angle of 315 degrees between each character's baseline and the x-axis of the device.

**Weight (4 bytes)**: 0x000000C8 specifies that the [**weight**](#gt_fc6de273-4e8b-4f97-89d0-3dc1b4be9e8d) of the font is 200, in the range 0 through 1000, from lightest to darkest, with 400 (0x00000190) considered normal.

**Italic (1 byte)**: 0x00 specifies that the font is not italic.

**Underline (1 byte)**: 0x00 specifies that the font is not underlined.

**Strikeout (1 byte)**: 0x00 specifies that the font characters do not have a strike-out graphic.

**CharSet (1 byte)**: 0x01 specifies the default character set, from the CharacterSet enumeration ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.1.1.5).

**OutPrecision (1 byte)**: 0x04 specifies the output precision, which is how closely the output is expected to match the requested font properties, from the OutPrecision enumeration ([MS-WMF] section 2.1.1.21). The value 0x04 specifies a [**TrueType**](#gt_855615c9-5378-459f-8d58-20f640afed6e) font, if there is a choice between multiple fonts with the same name.

**ClipPrecision (1 byte)**: 0x00 specifies the clipping precision, which is how to clip characters that are partially outside the clipping [**regions**](#gt_6f13fe5f-8747-4ae0-9375-814bf0528197), from the ClipPrecision flags ([MS-WMF] section 2.1.2.1). The value 0x00 specifies default clipping behavior.

**Quality (1 byte)**: 0x00 specifies default output quality, from the FontQuality enumeration ([MS-WMF] section 2.1.1.10).

**PitchAndFamily (1 byte)**: 0x02 specifies a variable-pitch font, and no preference for font family, from the PitchAndFamily object ([MS-WMF] section 2.2.2.14).

**Facename (68 bytes)**: "Arial" specifies the [**typeface**](#gt_2fff9b0b-4bc1-4d05-81ef-8dde59fd9789) name of the font in [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) characters.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| FullName ("") (132 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Style ("") (68 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Script ("") (68 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Signature (0x08007664) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NumAxes (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**FullName (132 bytes)**: An empty string specifies the font's full name.

**Style (68 bytes)**: An empty string describes the font's style.

**Script (68 bytes)**: An empty string describes the font's character set.

**Signature (4 bytes)**: 0x08007664 specifies the signature of a [DesignVector object (section 2.2.3)](#Section_cdf6681fbf454eec923e08c8be16aaf2).

**NumAxes (4 bytes)**: 0x00000000 specifies the number of [**font axes**](#gt_5d727201-0a47-44b1-86e5-faf77f0f2463) described in the DesignVector object.

### EMR\_SELECTOBJECT Example 3

This section provides an example of an [EMR\_SELECTOBJECT](#Section_145b063d5f9641feb7ae1e615b2bc2bf) record (section 2.3.8.5).

1. 00003280: 25 00 00 00
2. 00003290:0C 00 00 00 02 00 00 00

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type (0x00000025) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size (0x0000000C) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ihObject (0x00000002) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes)**: 0x00000025 identifies this record type as EMR\_SELECTOBJECT.

**Size (4 bytes)**: 0x0000000C specifies the size of this record in bytes.

**ihObject (4 bytes)**: 0x00000002 specifies the index of an object in the EMF object table.

### EMR\_EXTTEXTOUTW Example

This section provides an example of an [EMR\_EXTTEXTOUTW](#Section_a59a79ac328e492da34de02727af6edf) record (section 2.3.5.8).

1. 00003290: 54 00 00 00 A0 00 00 00
2. 000032A0:00 00 00 00 00 00 00 00 FF FF FF FF FF FF FF FF
3. 000032B0:01 00 00 00 AB 0A 0D 42 00 00 0D 42 12 00 00 00
4. 000032C0:05 00 00 00 0E 00 00 00 4C 00 00 00 00 00 00 00
5. 000032D0:00 00 00 00 00 00 00 00 FF FF FF FF FF FF FF FF
6. 000032E0:68 00 00 00 53 00 69 00 6D 00 70 00 6C 00 65 00
7. 000032F0:20 00 53 00 61 00 6D 00 70 00 6C 00 65 00 00 00
8. 00003300:09 00 00 00 03 00 00 00 0B 00 00 00 07 00 00 00
9. 00003310:03 00 00 00 07 00 00 00 04 00 00 00 09 00 00 00
10. 00003320:07 00 00 00 0B 00 00 00 07 00 00 00 03 00 00 00
11. 00003330:07 00 00 00 09 00 00 00

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type (0x00000054) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size (0x000000A0) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bounds (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... (0xFFFFFFFF) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... (0xFFFFFFFF) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| iGraphicsMode (0x00000001) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| exScale (35.260418) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| eyScale (35.250000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| aemrtext (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes)**: 0x00000054 identifies this record type as EMR\_EXTTEXTOUTW.

**Size (4 bytes)**: 0x000000A0 specifies the size of this record in bytes.

**Bounds (16 bytes)**: 0x00000000, 0x00000000, 0xFFFFFFFF, 0xFFFFFFFF values are not used.

**iGraphicsMode (4 bytes)**: 0x00000001 specifies the current graphics mode.

**exScale (4 bytes)**: 35.260418 specifies the X scale from page units to .01mm units if graphics mode is **GM\_COMPATIBLE**.

**eyScale (4 bytes)**: 35.250000 specifies the Y scale from page units to .01mm units if graphics mode is **GM\_COMPATIBLE**.

**aemrtext (variable)**: An array of [EmrText](#Section_dd585d0a5d7c4034963a1141af836972) objects (section 2.2.5) that specifies the properties of the strings to be output, and where to find the output strings and spacing values.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Reference (0x00000012) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... (0x00000005) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Chars (0x0000000E) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offString (0x0000004C) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Options (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Rectangle (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... (0xFFFFFFFF) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... (0xFFFFFFFF) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offDx (0x00000068) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| text ("Simple Sample") | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Reference (8 bytes)**: 0x00000012, 0x00000005 specify the coordinates of the reference point used to position the string.

**Chars (4 bytes)**: 0x0000000E specifies the number of characters in the string.

**offString (4 bytes)**: 0x0000004C specifies the offset to the string in bytes, from the start of the EMR\_EXTTEXTOUTW record.

**Options (4 bytes)**: 0x00000000 specifies that the **Rectangle** field is not used.

**Rectangle (16 bytes)**: 0x00000000, 0x00000000, 0xFFFFFFFF, 0xFFFFFFFF values are not used.

**offDx (4 bytes)**: 0x00000068 specifies the offset to an intercharacter spacing array in bytes, from the start of the EMR\_EXTTEXTOUTW record.

**text (4 bytes)**: "Simple Sample".

### EMR\_EXTCREATEFONTINDIRECTW Example 2

This section provides an example of an [EMR\_EXTCREATEFONTINDIRECTW](#Section_7e266b6d32e54201b6878ec40c24cd73) record (section 2.3.7.8).

1. 00003330: 52 00 00 00 70 01 00 00
2. 00003340:03 00 00 00 F3 FF FF FF 00 00 00 00 4E 0C 00 00
3. 00003350:4E 0C 00 00 C8 00 00 00 00 00 00 00 04 00 00 02
4. 00003360:4D 00 69 00 63 00 72 00 6F 00 73 00 6F 00 66 00
5. 00003370:74 00 20 00 53 00 61 00 6E 00 73 00 20 00 53 00
6. 00003380:65 00 72 00 69 00 66 00 00 00 00 00 00 00 00 00
7. 00003390:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8. 000033A0:00 00 00 00 28 00 00 00 59 00 00 00 2D 00 00 00
9. 000033B0:01 00 18 00 00 00 00 00 1C 2F 00 00 00 00 00 00
10. 000033C0:00 00 00 00 00 00 00 00 00 00 00 00 10 00 90 01
11. 000033D0:00 00 00 00 25 00 00 00 00 00 00 00 D3 3F EC FE
12. 000033E0:FE 07 00 00 79 0D 21 11 00 00 00 00 10 00 90 01
13. 000033F0:00 00 00 00 00 00 00 00 59 00 00 00 2D 00 00 00
14. 00003400:64 76 00 08 00 00 00 00 00 00 00 00 00 08 0D 00
15. 00003410:0D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16. 00003420:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
17. 00003430:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
18. 00003440:00 00 00 00 92 A0 CD 02 00 00 00 00 CA BE CD 02
19. 00003450:00 00 00 00 00 00 00 00 00 00 00 00 FF FF 5A FE
20. 00003460:00 00 00 00 00 00 00 00 00 00 00 00 95 F1 53 FE
21. 00003470:FE 07 00 00 BE 06 5A FE FE 07 00 00 87 F2 53 FE
22. 00003480:FE 07 00 00 C4 04 5A FE FE 07 00 00 79 0D 21 11
23. 00003490:00 00 00 00 01 00 00 00 00 00 00 00 F0 02 5A FE
24. 000034A0:64 76 00 08 00 00 00 00

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type (0x00000052) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size (0x00000170) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ihFonts (0x00000003) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| elw (360 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes)**: 0x00000052 identifies this record type as EMR\_EXTCREATEFONTINDIRECTW.

**Size (4 bytes)**: 0x00000170 specifies the size of this record in bytes.

**ihFonts (4 bytes)**: 0x00000003 specifies the object index that is assigned to the font.

**elw (360 bytes)**: To determine the type of logical font object in this field, an algorithm (section 2.3.7.8) is applied, which indicates this is a [LogFontExDv object (section 2.2.15)](#Section_595fcaaebc3741489afe859e81ca7f76).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Height (0xFFFFFFF3) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Width (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Escapement (0x00000C4E) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Orientation (0x00000C4E) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Weight 0x000000C8) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Italic (0x00) | | | | | | | | Underline (0x00) | | | | | | | | StrikeOut (0x00) | | | | | | | | CharSet (0x00) | | | | | | | |
| OutPrecision (0x04) | | | | | | | | ClipPrecision (0x00) | | | | | | | | Quality (0x00) | | | | | | | | PitchAndFamily (0x02) | | | | | | | |
| Facename ("Microsoft Sans Serif") (68 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Height (4 bytes)**: 0xFFFFFFF3 has an absolute value of 13, which specifies the character height for this font in logical units.

**Width (4 bytes)**: 0x00000000 specifies a computed font width. The aspect ratio of the device is matched against the digitization aspect ratio of the font to find the closest match, determined by the absolute value of the difference.

**Escapement (4 bytes)**: 0x00000C4E specifies an angle of 315 degrees between the [**baseline**](#gt_81ddcdfb-c66c-4c83-b963-18da9456fa62) of a row of text and the x-axis of the device.

**Orientation (4 bytes)**: 0x00000C4E specifies an angle of 315 degrees between each character's baseline and the x-axis of the device.

**Weight (4 bytes)**: 0x000000C8 specifies that the [**weight**](#gt_fc6de273-4e8b-4f97-89d0-3dc1b4be9e8d) of the font is 200, in the range 0 through 1000, from lightest to darkest, with 400 (0x00000190) considered normal.

**Italic (1 byte)**: 0x00 specifies that the font is not italic.

**Underline (1 byte)**: 0x00 specifies that the font is not underlined.

**Strikeout (1 byte)**: 0x00 specifies that the font characters do not have a strike-out graphic.

**CharSet (1 byte)**: 0x00 specifies the **ANSI\_CHARSET**, as defined in the CharacterSet enumeration ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.1.1.5).

**OutPrecision (1 byte)**: 0x04 specifies the output precision, which is how closely the output is expected to match the requested font properties, from the OutPrecision enumeration ([MS-WMF] section 2.1.1.21). The value 0x04 specifies a [**TrueType**](#gt_855615c9-5378-459f-8d58-20f640afed6e) font, if there is a choice between multiple fonts with the same name.

**ClipPrecision (1 byte)**: 0x00 specifies the clipping precision, which is how to clip characters that are partially outside the clipping [**region**](#gt_6f13fe5f-8747-4ae0-9375-814bf0528197), from the ClipPrecision flags ([MS-WMF] section 2.1.2.1). The value 0x00 specifies default clipping behavior.

**Quality (1 byte)**: 0x00 specifies default output quality, from the FontQuality enumeration ([MS-WMF] section 2.1.1.10).

**PitchAndFamily (1 byte)**: 0x02 specifies a variable-pitch font, and no preference for font family, from the PitchAndFamily object ([MS-WMF] section 2.2.2.14).

**Facename (68 bytes)**: "Microsoft Sans Serif" specifies the [**typeface**](#gt_2fff9b0b-4bc1-4d05-81ef-8dde59fd9789) name of the font in [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) characters.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| FullName ("") (132 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Style ("") (68 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Script ("") (68 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Signature (0x80007664) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NumAxes (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**FullName (132 bytes)**: An empty string specifies the font's full name.

**Style (68 bytes)**: An empty string describes the font's style.

**Script (68 bytes)**: An empty string describes the font's character set.

**Signature (4 bytes)**: 0x08007664 specifies the signature of a [DesignVector object (section 2.2.3)](#Section_cdf6681fbf454eec923e08c8be16aaf2).

**NumAxes (4 bytes)**: 0x00000000 specifies the number of [**font axes**](#gt_5d727201-0a47-44b1-86e5-faf77f0f2463) described in the DesignVector object.

### EMR\_SELECTOBJECT Example 4

This section provides an example of an [EMR\_SELECTOBJECT](#Section_145b063d5f9641feb7ae1e615b2bc2bf) record (section 2.3.8.5).

1. 000034A0: 25 00 00 00 0C 00 00 00
2. 000034B0:03 00 00 00

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type (0x00000025) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size (0x0000000C) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ihObject (0x00000003) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes)**: Identifies this record type as EMR\_SELECTOBJECT.

**Size (4 bytes)**: Specifies the size of this record in bytes.

**ihObject (4 bytes)**: Specifies the index of an object in the EMF object table.

### EMR\_EXTCREATEFONTINDIRECTW Example 3

This section provides an example of an EMR\_EXTCREATEFONTINDIRECTW record (section [2.3.7.8](#Section_7e266b6d32e54201b6878ec40c24cd73)).

1. 000034B0: 52 00 00 00 70 01 00 00 04 00 00 00
2. 000034C0:F2 FF FF FF 00 00 00 00 4E 0C 00 00 4E 0C 00 00
3. 000034D0:C8 00 00 00 00 00 00 00 04 00 00 02 4D 00 69 00
4. 000034E0:63 00 72 00 6F 00 73 00 6F 00 66 00 74 00 20 00
5. 000034F0:53 00 61 00 6E 00 73 00 20 00 53 00 65 00 72 00
6. 00003500:69 00 66 00 00 00 00 00 00 00 00 00 00 00 00 00
7. 00003510:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8. 00003520:C8 F1 13 00 00 00 00 00 10 00 90 01 00 00 00 00
9. 00003530:20 AC CF 02 00 00 00 00 10 00 00 00 06 00 00 00
10. 00003540:06 00 00 00 04 00 00 00 01 00 00 00 01 00 00 00
11. 00003550:01 00 00 00 01 00 00 00 0D 00 00 00 00 00 00 00
12. 00003560:03 00 00 00 00 08 00 00 3B 09 00 00 00 00 00 00
13. 00003570:A0 E8 07 02 00 00 00 00 03 00 00 00 FE 07 00 00
14. 00003580:90 01 00 00 4D 00 69 00 63 00 72 00 6F 00 73 00
15. 00003590:6F 00 66 00 74 00 20 00 53 00 61 00 00 00 73 00
16. 000035A0:20 00 53 00 65 00 72 00 69 00 66 00 00 00 00 00
17. 000035B0:00 00 00 00 00 00 00 00 FF FF 5A FE 00 00 00 00
18. 000035C0:40 02 5A FE FE 07 00 00 9D 04 00 00 00 00 00 00
19. 000035D0:FF FF FF FF FF FF FF FF 01 00 00 00 00 00 00 00
20. 000035E0:20 AC CF 02 00 00 00 00 00 00 07 02 00 00 00 00
21. 000035F0:10 AC CF 02 00 00 00 00 26 36 E3 76 00 00 00 00
22. 00003600:00 00 07 02 00 00 00 00 01 00 00 00 00 27 00 00
23. 00003610:00 00 00 00 00 00 00 00 20 AC CF 02 64 76 00 08
24. 00003620:00 00 00 00

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type (0x00000052) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size (0x00000170) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ihFonts (0x00000004) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| elw (360 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes)**: 0x00000052 identifies this record type as EMR\_EXTCREATEFONTINDIRECTW.

**Size (4 bytes)**: 0x00000170 specifies the size of this record in bytes.

**ihFonts (4 bytes)**: 0x00000004 specifies the object index that is assigned to the font.

**elw (360 bytes)**: To determine the type of logical font object in this field, an algorithm (section 2.3.7.8) is applied, which indicates this is a LogFontExDv object (section [2.2.15](#Section_595fcaaebc3741489afe859e81ca7f76)).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Height (0xFFFFFFF2) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Width (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Escapement (0x00000C4E) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Orientation (0x00000C4E) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Weight 0x000000C8) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Italic (0x00) | | | | | | | | Underline (0x00) | | | | | | | | StrikeOut (0x00) | | | | | | | | CharSet (0x00) | | | | | | | |
| OutPrecision (0x04) | | | | | | | | ClipPrecision (0x00) | | | | | | | | Quality (0x00) | | | | | | | | PitchAndFamily (0x02) | | | | | | | |
| Facename ("Microsoft Sans Serif") (68 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Height (4 bytes)**: 0xFFFFFFF2 has an absolute value of 14, which specifies the character height for this font in logical units.

**Width (4 bytes)**: 0x00000000 specifies a computed font width. The aspect ratio of the device is matched against the digitization aspect ratio of the font to find the closest match, determined by the absolute value of the difference.

**Escapement (4 bytes)**: 0x00000C4E specifies an angle of 315 degrees between the [**baseline**](#gt_81ddcdfb-c66c-4c83-b963-18da9456fa62) of a row of text and the x-axis of the device.

**Orientation (4 bytes)**: 0x00000C4E specifies an angle of 315 degrees between each character's baseline and the x-axis of the device.

**Weight (4 bytes)**: 0x000000C8 specifies that the [**weight**](#gt_fc6de273-4e8b-4f97-89d0-3dc1b4be9e8d) of the font is 200, in the range 0 through 1000, from lightest to darkest, with 400 (0x00000190) considered normal.

**Italic (1 byte)**: 0x00 specifies that the font is not italic.

**Underline (1 byte)**: 0x00 specifies that the font is not underlined.

**Strikeout (1 byte)**: 0x00 specifies that the font characters do not have a strike-out graphic.

**CharSet (1 byte)**: 0x00 specifies the **ANSI\_CHARSET**, as defined in the CharacterSet enumeration ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.1.1.5).

**OutPrecision (1 byte)**: 0x04 specifies the output precision, which is how closely the output is expected to match the requested font properties, from the OutPrecision enumeration ([MS-WMF] section 2.1.1.21). The value 0x04 specifies a [**TrueType**](#gt_855615c9-5378-459f-8d58-20f640afed6e) font, if there is a choice between multiple fonts with the same name.

**ClipPrecision (1 byte)**: 0x00 specifies the clipping precision, which is how to clip characters that are partially outside the clipping [**region**](#gt_6f13fe5f-8747-4ae0-9375-814bf0528197), from the ClipPrecision flags ([MS-WMF] section 2.1.2.1). The value 0x00 specifies default clipping behavior.

**Quality (1 byte)**: 0x00 specifies default output quality, from the FontQuality enumeration ([MS-WMF] section 2.1.1.10).

**PitchAndFamily (1 byte)**: 0x02 specifies a variable-pitch font, and no preference for font family, from the PitchAndFamily object ([MS-WMF] section 2.2.2.14).

**Facename (68 bytes)**: "Microsoft Sans Serif" specifies the [**typeface**](#gt_2fff9b0b-4bc1-4d05-81ef-8dde59fd9789) name of the font in [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) characters.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| FullName ("") (132 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Style ("") (68 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Script ("") (68 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Signature (0x80007664) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NumAxes (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**FullName (132 bytes)**: An empty string specifies the font's full name.

**Style (68 bytes)**: An empty string describes the font's style.

**Script (68 bytes)**: An empty string describes the font's character set.

**Signature (4 bytes)**: 0x08007664 specifies the signature of a [DesignVector object (section 2.2.3)](#Section_cdf6681fbf454eec923e08c8be16aaf2).

**NumAxes (4 bytes)**: 0x00000000 specifies the number of [**font axes**](#gt_5d727201-0a47-44b1-86e5-faf77f0f2463) described in the DesignVector object.

### EMR\_SELECTOBJECT Example 5

This section provides an example of an [EMR\_SELECTOBJECT](#Section_145b063d5f9641feb7ae1e615b2bc2bf) record (section 2.3.8.5).

1. 00003620: 25 00 00 00 0C 00 00 00 04 00 00 00

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type (0x00000025) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size (0x0000000C) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ihObject (0x00000004) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes)**: Identifies this record type as EMR\_SELECTOBJECT.

**Size (4 bytes)**: Specifies the size of this record in bytes.

**ihObject (4 bytes)**: Specifies the index of an object in the EMF object table.

### EMR\_DELETEOBJECT Example 1

This section provides an example of an [EMR\_DELETEOBJECT](#Section_6f0f12a3111a478b8251a9505168f9a9) record (section 2.3.8.3).

1. 00003630:28 00 00 00 0C 00 00 00 03 00 00 00

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type (0x00000028) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size (0x0000000C) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ihObject (0x00000003) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes)**: Identifies this record type as EMR\_DELETEOBJECT.

**Size (4 bytes)**: Specifies the size of this record in bytes.

**ihObject (4 bytes)**: Specifies the index of the object to be deleted.

### EMR\_EXTCREATEFONTINDIRECTW Example 4

This section provides an example of an [EMR\_EXTCREATEFONTINDIRECTW](#Section_7e266b6d32e54201b6878ec40c24cd73) record (section 2.3.7.8).

1. 00003630: 52 00 00 00
2. 00003640:70 01 00 00 03 00 00 00 13 00 00 00 00 00 00 00
3. 00003650:4E 0C 00 00 4E 0C 00 00 C8 00 00 00 00 00 00 00
4. 00003660:04 00 00 02 4D 00 69 00 63 00 72 00 6F 00 73 00
5. 00003670:6F 00 66 00 74 00 20 00 53 00 61 00 6E 00 73 00
6. 00003680:20 00 53 00 65 00 72 00 69 00 66 00 00 00 00 00
7. 00003690:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8. 000036A0:00 00 00 00 00 00 73 00 20 00 53 00 65 00 72 00
9. 000036B0:69 00 66 00 00 00 00 00 00 00 00 00 00 00 00 00
10. 000036C0:FF FF 5A FE 00 00 00 00 40 02 5A FE FE 07 00 00
11. 000036D0:9D 04 00 00 00 00 00 00 FF FF FF FF FF FF FF FF
12. 000036E0:01 00 00 00 00 00 00 00 20 AC CF 02 00 00 00 00
13. 000036F0:00 00 07 02 00 00 00 00 40 02 5A FE FE 07 00 00
14. 00003700:F3 14 00 00 00 00 00 00 F3 14 0A 1E 00 00 00 00
15. 00003710:94 8A E8 FE FE 07 00 00 04 00 00 00 00 00 00 00
16. 00003720:65 58 53 FE 00 00 00 00 00 00 00 00 00 00 00 00
17. 00003730:00 F5 13 00 00 00 00 00 03 01 56 E5 89 1A 00 00
18. 00003740:55 00 00 00 00 00 00 00 00 00 00 00 04 00 00 00
19. 00003750:00 00 00 00 FE 07 00 00 79 0D 21 11 00 00 00 00
20. 00003760:40 02 5A FE 00 00 00 00 26 06 5A FE FE 07 00 00
21. 00003770:08 F5 13 00 00 00 00 00 00 F5 13 00 00 00 00 00
22. 00003780:07 CB 54 FE FE 07 00 00 79 0D 21 11 00 00 00 00
23. 00003790:04 00 00 00 00 00 00 00 24 07 5A FE FE 07 00 00
24. 000037A0:01 00 00 00 64 76 00 08 00 00 00 00

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type (0x00000052) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size (0x00000170) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ihFonts (0x00000003) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| elw (360 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes)**: 0x00000052 identifies this record type as EMR\_EXTCREATEFONTINDIRECTW.

**Size (4 bytes)**: 0x00000170 specifies the size of this record in bytes.

**ihFonts (4 bytes)**: 0x00000003 specifies the object index that is assigned to the font.

**elw (360 bytes)**: To determine the type of logical font object in this field, an algorithm (section 2.3.7.8) is applied, which indicates this is a [LogFontExDv object (section 2.2.15)](#Section_595fcaaebc3741489afe859e81ca7f76).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Height (0x00000013) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Width (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Escapement (0x00000C4E) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Orientation (0x00000C4E) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Weight 0x000000C8) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Italic (0x00) | | | | | | | | Underline (0x00) | | | | | | | | StrikeOut (0x00) | | | | | | | | CharSet (0x01) | | | | | | | |
| OutPrecision (0x04) | | | | | | | | ClipPrecision (0x00) | | | | | | | | Quality (0x00) | | | | | | | | PitchAndFamily (0x02) | | | | | | | |
| Facename ("Microsoft Sans Serif") (68 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Height (4 bytes)**: 0x00000013 specifies the [**cell height**](#gt_b97118e3-2175-4af5-b1f9-462e57501090) for this font, in logical units.

**Width (4 bytes)**: 0x00000000 specifies a computed font width. The [**aspect ratio**](#gt_091ff8ed-6ac1-4015-9d91-93020f13f6b5) of the device is matched against the digitization aspect ratio of the font to find the closest match, determined by the absolute value of the difference.

**Escapement (4 bytes)**: 0x00000C4E specifies an angle of 315 degrees between the [**baseline**](#gt_81ddcdfb-c66c-4c83-b963-18da9456fa62) of a row of text and the x-axis of the device.

**Orientation (4 bytes)**: 0x00000C4E specifies an angle of 315 degrees between each character's baseline and the x-axis of the device.

**Weight (4 bytes)**: 0x000000C8 specifies that the [**weight**](#gt_fc6de273-4e8b-4f97-89d0-3dc1b4be9e8d) of the font is 200, in the range 0 through 1000, from lightest to darkest, with 400 (0x00000190) considered normal.

**Italic (1 byte)**: 0x00 specifies that the font is not italic.

**Underline (1 byte)**: 0x00 specifies that the font is not underlined.

**Strikeout (1 byte)**: 0x00 specifies that the font characters do not have a strike-out graphic.

**CharSet (1 byte)**: 0x00 specifies the **ANSI\_CHARSET**, as defined in the CharacterSet enumeration ([[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) section 2.1.1.5).

**OutPrecision (1 byte)**: 0x04 specifies the output precision, which is how closely the output is expected to match the requested font properties, from the OutPrecision enumeration ([MS-WMF] section 2.1.1.21). The value 0x04 specifies a [**TrueType**](#gt_855615c9-5378-459f-8d58-20f640afed6e) font, if there is a choice between multiple fonts with the same name.

**ClipPrecision (1 byte)**: 0x00 specifies the clipping precision, which is how to clip characters that are partially outside the clipping [**region**](#gt_6f13fe5f-8747-4ae0-9375-814bf0528197), from the ClipPrecision flags ([MS-WMF] section 2.1.2.1). The value 0x00 specifies default clipping behavior.

**Quality (1 byte)**: 0x00 specifies default output quality, from the FontQuality enumeration ([MS-WMF] section 2.1.1.10).

**PitchAndFamily (1 byte)**: 0x02 specifies a variable-pitch font, and no preference for font family, from the PitchAndFamily object ([MS-WMF] section 2.2.2.14).

**Facename (68 bytes)**: "Microsoft Sans Serif" specifies the [**typeface**](#gt_2fff9b0b-4bc1-4d05-81ef-8dde59fd9789) name of the font in [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) characters.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| FullName ("") (132 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Style ("") (68 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Script ("") (68 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Signature (0x80007664) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NumAxes (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**FullName (132 bytes)**: An empty string specifies the font's full name.

**Style (68 bytes)**: An empty string describes the font's style.

**Script (68 bytes)**: An empty string describes the font's character set.

**Signature (4 bytes)**: 0x08007664 specifies the signature of a [DesignVector object (section 2.2.3)](#Section_cdf6681fbf454eec923e08c8be16aaf2).

**NumAxes (4 bytes)**: 0x00000000 specifies the number of [**font axes**](#gt_5d727201-0a47-44b1-86e5-faf77f0f2463) described in the DesignVector object.

### EMR\_SELECTOBJECT Example 6

This section provides an example of an [EMR\_SELECTOBJECT](#Section_145b063d5f9641feb7ae1e615b2bc2bf) record (section 2.3.8.5).

1. 000037A0: 25 00 00 00
2. 000037B0:0c 00 00 00 03 00 00 00

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type (0x00000025) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size (0x0000000C) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ihObject (0x00000003) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes)**: 0x00000025 identifies this record type as EMR\_SELECTOBJECT.

**Size (4 bytes)**: 0x0000000C specifies the size of this record in bytes.

**ihObject (4 bytes)**: 0x00000003 specifies the index of an object in the EMF object table.

### EMR\_SELECTOBJECT Example 7

This section provides an example of an [EMR\_SELECTOBJECT](#Section_145b063d5f9641feb7ae1e615b2bc2bf) record (section 2.3.8.5).

1. 000037B0: 25 00 00 00 0C 00 00 00
2. 000037C0:02 00 00 00

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type (0x00000025) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size (0x0000000C) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ihObject (0x00000002) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes)**: 0x00000025 identifies this record type as EMR\_SELECTOBJECT.

**Size (4 bytes)**: 0x0000000C specifies the size of this record in bytes.

**ihObject (4 bytes)**: 0x00000002 specifies the index of an object in the EMF object table.

### EMR\_DELETEOBJECT Example 2

This section provides an example of an [EMR\_DELETEOBJECT](#Section_6f0f12a3111a478b8251a9505168f9a9) record (section 2.3.8.3).

1. 000037C0: 28 00 00 00 0C 00 00 00 04 00 00 00

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type (0x00000028) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size (0x0000000C) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ihObject (0x00000004) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes)**: 0x00000028 identifies this record type as EMR\_DELETEOBJECT.

**Size (4 bytes)**: 0x0000000C specifies the size of this record in bytes.

**ihObject (4 bytes)**: 0x00000004 specifies the index of the object to be deleted.

### EMR\_DELETEOBJECT Example 3

This section provides an example of an [EMR\_DELETEOBJECT](#Section_6f0f12a3111a478b8251a9505168f9a9) record (section 2.3.8.3).

1. 000037D0:28 00 00 00 0C 00 00 00 03 00 00 00

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type (0x00000028) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size (0x0000000C) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ihObject (0x00000003) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes)**: 0x00000028 identifies this record type as EMR\_DELETEOBJECT.

**Size (4 bytes)**: 0x0000000C specifies the size of this record in bytes.

**ihObject (4 bytes)**: 0x00000003 specifies the index of the object to be deleted.

### EMR\_SELECTOBJECT Example 8

This section provides an example of an [EMR\_SELECTOBJECT](#Section_145b063d5f9641feb7ae1e615b2bc2bf) record (section 2.3.8.5).

1. 000037D0: 25 00 00 00
2. 000037E0:0c 00 00 00 0D 00 00 80

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type (0x00000025) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size (0x0000000C) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ihObject (0x8000000D = SYSTEM\_FONT) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes)**: 0x00000025 identifies this record type as EMR\_SELECTOBJECT.

**Size (4 bytes)**: 0x0000000C specifies the size of this record in bytes.

**ihObject (4 bytes)**: 0x8000000D specifies the index of an object in the EMF object table.

### EMR\_EOF Example

This section provides an example of an [EMR\_EOF](#Section_3f47fde00e6b40c187f3f4129af03aa1) record (section 2.3.4.1).

1. 000037E0: 0E 00 00 00 14 00 00 00
2. 000037F0:00 00 00 00 10 00 00 00 14 00 00 00

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Type (0x0000000E) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size (0x00000014) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| nPalEntries (0x00000000) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| offPalEntries (0x00000010) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SizeLast (0x00000014) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type (4 bytes)**: 0x0000000E identifies this record type as EMR\_EOF.

**Size (4 bytes)**: 0x00000014 specifies the size of this record in bytes.

**nPalEntries (4 bytes)**: 0x00000000 specifies the number of [**palette**](#gt_dca5cd91-8d01-424b-9722-c9e818562ee5) entries.

**offPalEntries (4 bytes)**: 0x00000010 specifies the offset to the palette entries.

**SizeLast (4 bytes)**: 0x00000014 is the same as the **Size** value. It is the last field in the [**metafile**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7).

# Security Considerations

This file format enables third parties to send payloads (such as [**PostScript**](#gt_31971c63-f974-47de-8de9-6b76d093890d)) to pass through as executable code.

# Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include updates to those products.

* Windows NT 3.1 operating system
* Windows NT 3.5 operating system
* Windows NT 3.51 operating system
* Windows NT 4.0 operating system
* Windows 98 operating system
* Windows Millennium Edition operating system
* Windows 2000 operating system
* Windows XP operating system
* Windows Server 2003 operating system
* Windows Vista operating system
* Windows Server 2008 operating system
* Windows 7 operating system
* Windows Server 2008 R2 operating system
* Windows 8 operating system
* Windows Server 2012 operating system
* Windows 8.1 operating system
* Windows Server 2012 R2 operating system
* Windows 10 operating system
* Windows Server 2016 operating system
* Windows Server operating system
* Windows Server 2019 operating system
* Windows Server 2022 operating system
* Windows 11 operating system

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms "SHOULD" or "SHOULD NOT" implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term "MAY" implies that the product does not follow the prescription.

[<1> Section 1.3.1](#Appendix_A_Target_1): **[EMF](#gt_d9d0bff9-d270-4528-9081-fe51db809c36)** [**metafiles**](#gt_ae5f028e-7e28-4a0b-bec6-2c87913f7db7) have changed over the years with the evolution of Windows operating systems. They were introduced with 32-bit Windows and replaced the 16-bit [**WMF**](#gt_48849cf6-d55c-47e5-b041-13b65854de2b) metafile [[MS-WMF]](%5bMS-WMF%5d.pdf#Section_4813e7fd52d04f42965f228c8b7488d2) as the Windows standard.

[<2> Section 1.4](#Appendix_A_Target_2): Windows applications that use the [**GDI+**](#gt_aad222f8-e733-44f2-a97c-1f903af82448) API [[MSDN-GDI+]](https://go.microsoft.com/fwlink/?LinkId=90013) can create EMF metafiles that contain [**EMF+**](#gt_41a5b58b-b050-4001-85ec-d6ed56379ee2) records ([[MS-EMFPLUS]](%5bMS-EMFPLUS%5d.pdf#Section_5f92c78964f246b59ed415a9bb0946c6) section 2.3).

[<3> Section 2.1.1](#Appendix_A_Target_3): Windows NT 3.1 is the only Windows version in which [**GDI**](#gt_6c550e84-9a9c-46ef-8716-7260fe29cae3) uses [EMR\_POLYTEXTOUTA](#Section_d8b7ac2176b84f9aa7cc05f9d2d5627e) records for text output. All other versions emulate EMR\_POLYTEXTOUTA with [EMR\_EXTTEXTOUTW](#Section_a59a79ac328e492da34de02727af6edf) records.

[<4> Section 2.1.1](#Appendix_A_Target_4): Windows NT 3.1 is the only Windows version in which GDI uses EMR\_POLYTEXTOUTW records for text output. All other versions emulate EMR\_POLYTEXTOUTW with EMR\_EXTTEXTOUTW records.

[<5> Section 2.1.1](#Appendix_A_Target_5): Windows NT 3.1, Windows NT 3.5, and Windows NT 3.51 do not support EMR\_SETICMMODE.

[<6> Section 2.1.1](#Appendix_A_Target_6): Windows NT 3.1, Windows NT 3.5, and Windows NT 3.51 do not support EMR\_CREATECOLORSPACE.

[<7> Section 2.1.1](#Appendix_A_Target_7): Windows NT 3.1, Windows NT 3.5, and Windows NT 3.51 do not support EMR\_SETCOLORSPACE.

[<8> Section 2.1.1](#Appendix_A_Target_8): Windows NT 3.1, Windows NT 3.5, and Windows NT 3.51 do not support EMR\_DELETECOLORSPACE.

[<9> Section 2.1.1](#Appendix_A_Target_9): Windows uses an [EMR\_DELETEOBJECT](#Section_6f0f12a3111a478b8251a9505168f9a9) record to delete a logical color space object.

[<10> Section 2.1.1](#Appendix_A_Target_10): Windows NT 3.1, Windows NT 3.5, and Windows NT 3.51 do not support EMR\_GLSRECORD.

[<11> Section 2.1.1](#Appendix_A_Target_11): Windows NT 3.1, Windows NT 3.5, and Windows NT 3.51 do not support EMR\_GLSBOUNDEDRECORD.

[<12> Section 2.1.1](#Appendix_A_Target_12): Windows NT 3.1, Windows NT 3.5, and Windows NT 3.51 do not support EMR\_PIXELFORMAT.

[<13> Section 2.1.1](#Appendix_A_Target_13): Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, and Windows NT 4.0 do not support **EMR\_COLORCORRECTPALETTE**.

[<14> Section 2.1.1](#Appendix_A_Target_14): Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, and Windows NT 4.0 do not support **EMR\_SETICMPROFILEA**.

[<15> Section 2.1.1](#Appendix_A_Target_15): Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, and Windows NT 4.0 do not support **EMR\_SETICMPROFILEW**.

[<16> Section 2.1.1](#Appendix_A_Target_16): Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, and Windows NT 4.0 do not support **EMR\_ALPHABLEND**.

[<17> Section 2.1.1](#Appendix_A_Target_17): Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, and Windows NT 4.0 do not support **EMR\_SETLAYOUT**.

[<18> Section 2.1.1](#Appendix_A_Target_18): Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, and Windows NT 4.0 do not support **EMR\_TRANSPARENTBLT**.

[<19> Section 2.1.1](#Appendix_A_Target_19): Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, and Windows NT 4.0 do not support **EMR\_GRADIENTFILL**.

[<20> Section 2.1.1](#Appendix_A_Target_20): Windows GDI uses an EMR\_EXTTEXTOUTW record (section 2.3.5.8) to perform this function.

[<21> Section 2.1.1](#Appendix_A_Target_21): Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, and Windows NT 4.0 do not support **EMR\_COLORMATCHTOTARGETW**.

[<22> Section 2.1.1](#Appendix_A_Target_22): Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, and Windows NT 4.0 do not support **EMR\_CREATECOLORSPACEW**.

[<23> Section 2.1.11](#Appendix_A_Target_23): Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, Windows 98, and Windows Millennium Edition: Do not support the **ETO\_GLYPH\_INDEX** flag used for bitmap and [**vector fonts**](#gt_4795d9b6-f3e9-4435-874e-bb6882ea1f0f)—in addition to [**TrueType**](#gt_855615c9-5378-459f-8d58-20f640afed6e) fonts—to indicate that no further language processing is necessary and that GDI processes the string directly. See [MSDN-GDI+] for more information.

[<24> Section 2.1.11](#Appendix_A_Target_24): Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, Windows 98, and Windows Millennium Edition: Do not support the **ETO\_RTLREADING** flag used to indicate right-to-left reading order.

[<25> Section 2.1.11](#Appendix_A_Target_25): Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, Windows 98, and Windows Millennium Edition: Do not support the **ETO\_NUMERICSLOCAL** flag used to indicate the display of numeric digits appropriate to the locale.

[<26> Section 2.1.11](#Appendix_A_Target_26): Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, Windows 98, and Windows Millennium Edition: Do not support the **ETO\_NUMERICSLATIN** flag used to indicate the display of numeric digits appropriate to Europe.

[<27> Section 2.1.11](#Appendix_A_Target_27): Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, Windows 98, and Windows Millennium Edition: Do not support the **ETO\_IGNORELANGUAGE** flag used to indicate that international scripting support is not used, which might cause no text to be output.

[<28> Section 2.1.11](#Appendix_A_Target_28): Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, Windows 98, Windows Millennium Edition, Windows NT 4.0, and Windows 2000: Do not support the **ETO\_PDY** flag used to indicate that both horizontal and vertical character displacement values are provided.

[<29> Section 2.1.11](#Appendix_A_Target_29): Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, Windows 98, Windows Millennium Edition, Windows NT 4.0, and Windows 2000: Do not support the **ETO\_REVERSE\_INDEX\_MAP** flag.

[<30> Section 2.1.16](#Appendix_A_Target_30): **GM\_COMPATIBLE** graphics mode is used for compatibility between 16-bit and 32-bit systems.

[<31> Section 2.1.16](#Appendix_A_Target_31): Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, Windows 98, Windows Millennium Edition, Windows NT 4.0, and Windows 2000: **GM\_ADVANCED** is not supported.

[<32> Section 2.1.18](#Appendix_A_Target_32): Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, Windows 98, and Windows Millennium Edition: Do not support [**Image Color Management (ICM)**](#gt_fcd5ee55-57ff-45d6-b51e-2ccba47329e3).

[<33> Section 2.1.31](#Appendix_A_Target_33): On Windows, this is the "Courier" font.

[<34> Section 2.1.31](#Appendix_A_Target_34): On Windows, this is the "MS Sans Serif" font.

[<35> Section 2.1.31](#Appendix_A_Target_35): On Windows, this is the "Tahoma" font and is used to draw menu text and dialog box controls.

Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, Windows NT 4.0, Windows 98, and Windows Millennium Edition: The system font is "MS Sans Serif".

[<36> Section 2.1.31](#Appendix_A_Target_36): On Windows, this value is considered equivalent to SYSTEM\_FONT for the purposes of the screen display of metafiles.

Windows 98 and Windows Millennium Edition: This value is not supported.

[<37> Section 2.1.31](#Appendix_A_Target_37): On Windows, this [**palette**](#gt_dca5cd91-8d01-424b-9722-c9e818562ee5) consists of the static colors in the [**system palette**](#gt_874d5ba8-6fd5-4c63-97f3-f35cfeda95fe).

[<38> Section 2.1.31](#Appendix_A_Target_38): On Windows, the default user interface font is "Tahoma".

Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, Windows NT 4.0, Windows 98, and Windows Millennium Edition: The default user interface font is "MS Sans Serif".

[<39> Section 2.1.31](#Appendix_A_Target_39): Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, Windows NT 4.0, Windows 98, and Windows Millennium Edition: The default brush is undefined.

[<40> Section 2.1.31](#Appendix_A_Target_40): Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, Windows NT 4.0, Windows 98, and Windows Millennium Edition: The default pen is undefined.

[<41> Section 2.1.32](#Appendix_A_Target_41): Windows also uses the following symbolic names for the StretchMode enumeration; their meanings are exactly the same as the members with the same values.

1. #define BLACKONWHITE 1
2. #define WHITEONBLACK 2
3. #define COLORONCOLOR 3
4. #define HALFTONE 4

[<42> Section 2.2.2](#Appendix_A_Target_42): Windows 98 and Windows Millennium Edition do not support the [ColorAdjustment](#Section_71d7f4d0b6454de8a5588a2e0fc596cb) object.

[<43> Section 2.2.2](#Appendix_A_Target_43): Windows can generate ColorAdjustment objects with values outside their valid ranges. Such objects are ignored.

[<44> Section 2.2.5](#Appendix_A_Target_44): In Windows implementations, this is the clipping and/or opaquing rectangle that is passed to GDI methods **ExtTextOutA** and **ExtTextOutW**.

[<45> Section 2.2.6](#Appendix_A_Target_45): Windows does not parse the [**PostScript**](#gt_31971c63-f974-47de-8de9-6b76d093890d) data in an [EpsData object](#Section_a624b402f7aa4cb6b289612c9775f37e); the data is handed off to the graphics [**printer driver**](#gt_1a48eebd-e72c-494d-b8cb-84dfb7bc3b65) if the driver supports PostScript printing.

[<46> Section 2.2.13](#Appendix_A_Target_46): In Windows implementations, the [**aspect ratio**](#gt_091ff8ed-6ac1-4015-9d91-93020f13f6b5) of the device is matched against the digitization aspect ratios of the available fonts to find the closest match, determined by the absolute value of the difference.

[<47> Section 2.2.13](#Appendix_A_Target_47): Windows uses a [**weight**](#gt_fc6de273-4e8b-4f97-89d0-3dc1b4be9e8d) value of 400 by default.

| Value | Weight |
| --- | --- |
| Thin | 100 |
| Extra Light (Ultra Light) | 200 |
| Light | 300 |
| Normal (Regular) | 400 |
| Medium | 500 |
| Semi-Bold (Demi-Bold) | 600 |
| Bold | 700 |
| Extra Bold (Ultra Bold) | 800 |
| Heavy (Black) | 900 |

[<48> Section 2.2.22](#Appendix_A_Target_48): Windows implementations do not support this flag.

[<49> Section 2.2.22](#Appendix_A_Target_49): Windows can use this flag to indicate that the pixel format specified by this structure is supported by GDI. See [MSDN-GDI+] for more information.

Windows can also use this flag to specify single-buffering for the pixel buffer.

[<50> Section 2.2.22](#Appendix_A_Target_50): Windows uses this flag to indicate that the pixel pixel format is supported by GDI.

[<51> Section 2.2.22](#Appendix_A_Target_51): Windows uses this with [**OpenGL**](#gt_216becb9-a800-4a44-8206-2ec2231c9c67) drawing only.

Windows NT 3.1, Windows NT 3.51, Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, and Windows Server 2003 do not support this flag.

[<52> Section 2.2.22](#Appendix_A_Target_52): Windows does not support [**alpha**](#gt_134d5688-b046-4a5b-a74e-c0c11fc04eb3) bitplanes.

[<53> Section 2.2.22](#Appendix_A_Target_53): Windows does not support alpha bitplanes.

[<54> Section 2.2.22](#Appendix_A_Target_54): Windows does not support alpha bitplanes.

[<55> Section 2.2.27](#Appendix_A_Target_55): In this case, Windows uses the logical font that is currently selected in the [**playback device context**](#gt_32591a2b-a9d0-4ccf-a5b8-7177e1ea8d45).

[<56> Section 2.3.1.1](#Appendix_A_Target_56): Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, and Windows NT 4.0 do not support **EMR\_ALPHABLEND**.

[<57> Section 2.3.1.3](#Appendix_A_Target_57): Windows might set this to a non-zero value.

[<58> Section 2.3.1.8](#Appendix_A_Target_58): Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, and Windows NT 4.0 do not support **EMR\_TRANSPARENTBLT**.

[<59> Section 2.3.1.8](#Appendix_A_Target_59): Windows uses the [EMR\_ALPHABLEND](#Section_34e07d4faee64b63a4bb96996ad47669) record (section 2.3.1.1) to specify a block transfer of a 32 bits-per-pixel [**bitmap**](#gt_26a17e58-eeba-4210-81b9-8bbadf7ee952) with alpha transparency.

[<60> Section 2.3.3.1](#Appendix_A_Target_60): Windows NT 3.1, Windows NT 3.51, and Windows NT 4.0 ignore EMR\_COMMENT records.

[<61> Section 2.3.3.4.3](#Appendix_A_Target_61): On playback, the first graphics format recognized by Windows is used to render the image.

[<62> Section 2.3.4.2](#Appendix_A_Target_62): Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, Windows 98, and Windows Millennium Edition: EMF header extension 1 is not supported.

[<63> Section 2.3.4.2](#Appendix_A_Target_63): Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, Windows 98, Windows Millennium Edition, Windows NT 4.0, and Windows 2000: EMF header extension 2 is not supported.

[<64> Section 2.3.5.7](#Appendix_A_Target_64): Windows GDI emulates [EMR\_EXTTEXTOUTA](#Section_6b582a713c294fc6a0f41f8a313739a1) with an EMR\_EXTTEXTOUTW record.

[<65> Section 2.3.5.10](#Appendix_A_Target_65): The Windows playback implementation computes the bounding [**region**](#gt_6f13fe5f-8747-4ae0-9375-814bf0528197) from the sum of all the rectangles specified by the **RegionData** object in the **RgnData** field.

[<66> Section 2.3.5.11](#Appendix_A_Target_66): The Windows playback implementation computes the bounding region from the sum of all the rectangles specified by **RgnData** field.

[<67> Section 2.3.5.12](#Appendix_A_Target_67): Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, and Windows NT 4.0 do not support EMR\_GRADIENTFILL.

[<68> Section 2.3.5.12](#Appendix_A_Target_68): Windows uses true colors in 24-bits-per-pixel (bpp) and 32-bpp formats, and [**dithering**](#gt_0540a76e-7353-4f83-a800-377d1fb00903) in 4-bpp, 8-bpp, and 16-bpp formats.

[<69> Section 2.3.5.14](#Appendix_A_Target_69): The Windows playback implementation computes the bounding region from the sum of all the rectangles specified by the **RgnData** field.

[<70> Section 2.3.5.32](#Appendix_A_Target_70): Windows NT 3.1 is the only Windows version in which GDI uses EMR\_POLYTEXTOUTA records for text output. All other versions emulate EMR\_POLYTEXTOUTA with EMR\_EXTTEXTOUTW records.

[<71> Section 2.3.5.33](#Appendix_A_Target_71): Windows NT 3.1 is the only Windows version in which GDI uses [EMR\_POLYTEXTOUTW](#Section_381015eb29a7477493ada210697f5972) records for text output. All other versions emulate EMR\_POLYTEXTOUTW with EMR\_EXTTEXTOUTW records.

[<72> Section 2.3.7.2](#Appendix_A_Target_72): Windows NT 3.1, Windows NT 3.5, and Windows NT 3.51 do not support **EMR\_CREATECOLORSPACE**.

[<73> Section 2.3.7.3](#Appendix_A_Target_73): Windows NT 3.1, Windows NT 3.5, and Windows NT 3.51 do not support **EMR\_CREATECOLORSPACEW**.

[<74> Section 2.3.7.8](#Appendix_A_Target_74): Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, and Windows NT 4.0 metafiles contain a LogFont object in this field.

[<75> Section 2.3.7.8](#Appendix_A_Target_75): On Microsoft Windows, if the size of the **elw** field is less than the size of a **LogFontPanose** object, Windows GDI copies the data from the **elw** field into a zero-initialized **LogFontPanose** object and processes this new **LogFontPanose** object as input instead.

[<76> Section 2.3.8.1](#Appendix_A_Target_76): Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, and Windows NT 4.0 do not support **EMR\_COLORCORRECTPALETTE**.

[<77> Section 2.3.8.2](#Appendix_A_Target_77): Windows NT 3.1, Windows NT 3.5, and Windows NT 3.51 do not support **EMR\_DELETECOLORSPACE**.

[<78> Section 2.3.8.4](#Appendix_A_Target_78): Windows GDI does not perform parameter validation on this value, which can lead to the generation of EMF metafiles that contain invalid EMR\_RESIZEPALETTE records. Windows ignores such invalid records when processing metafiles.

[<79> Section 2.3.8.7](#Appendix_A_Target_79): Windows NT 3.1, Windows NT 3.5, and Windows NT 3.51 do not support **EMR\_SETCOLORSPACE**.

[<80> Section 2.3.9](#Appendix_A_Target_80): Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, Windows 98, and Windows Millennium Edition: OpenGL records are not supported.

[<81> Section 2.3.11.1](#Appendix_A_Target_81): Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, and Windows NT 4.0 do not support **EMR\_COLORMATCHTOTARGETW**.

[<82> Section 2.3.11.1](#Appendix_A_Target_82): On Windows NT 4.0, Windows 2000, Windows XP, and Windows Server 2003, [**WCS**](#gt_e4f9c4a4-c195-43d0-b189-f9727ec8a2b3) is not enabled in the playback device context before applying the current color transform.

[<83> Section 2.3.11.3](#Appendix_A_Target_83): The Windows playback implementation computes the bounding rectangle of the region from the sum of all the rectangles specified by the **RegionData** object.

[<84> Section 2.3.11.5](#Appendix_A_Target_84): Windows NT 3.1, Windows NT 3.5, and Windows NT 3.51 do not support **EMR\_PIXELFORMAT**.

[<85> Section 2.3.11.14](#Appendix_A_Target_85): Windows NT 3.1, Windows NT 3.5, and Windows NT 3.51 do not support **EMR\_SETICMMODE**.

[<86> Section 2.3.11.15](#Appendix_A_Target_86): Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, and Windows NT 4.0 do not support **EMR\_SETICMPROFILEA**.

[<87> Section 2.3.11.16](#Appendix_A_Target_87): Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, and Windows NT 4.0 do not support **EMR\_SETICMPROFILEW**.

[<88> Section 2.3.11.17](#Appendix_A_Target_88): Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, and Windows NT 4.0 do not support **EMR\_SETLAYOUT**.

[<89> Section 2.3.11.20](#Appendix_A_Target_89): Only Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, and Windows NT 4.0 support **EMR\_SETMAPPERFLAGS**.

[<90> Section 2.3.11.21](#Appendix_A_Target_90): Windows GDI accepts a FLOAT value for the [**miter length**](#gt_7d78ab9f-9531-4993-b140-8c2bb6e82220) limit value.

[<91> Section 2.3.11.27](#Appendix_A_Target_91): Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, Windows NT 4.0, Windows 98, and Windows Millennium Edition do not support the EMR\_SETTEXTJUSTIFICATION record type.

[<92> Section 3.1.1.2.1](#Appendix_A_Target_92): During metafile playback, before a clipping region is defined an EMF record, Windows uses a default based on the type of [**device context**](#gt_92e112c6-791f-48dd-8345-e65a35eef388). in the following table, the referenced functions are described in [[MSDN-WindowsGDI]](https://go.microsoft.com/fwlink/?LinkId=94735) unless stated otherwise.

| Device context type | Created by | Default clipping region |
| --- | --- | --- |
| Window DC | GetDC | Area of the window passed to GetDC |
| Memory DC compatible with a specified output device | CreateCompatibleDC with  CreateCompatibleBitmap | Area of the compatible bitmap |
| Monitor DC | MonitorEnumProc from  EnumDisplayMonitors | Area of the display monitor |
| Printer DC | CreateDC | (**dmPelsWidth** X **dmPelsHeight**) of the DEVMODE structure [[DEVMODE]](https://go.microsoft.com/fwlink/?LinkId=89844) input to CreateDC |
| Metafile DC | CreateMetaFile or CreateEnhMetaFile | None |

# Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as Major, Minor, or None.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

* A document revision that incorporates changes to interoperability requirements.
* A document revision that captures changes to protocol functionality.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **None** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the relevant technical content is identical to the last released version.

The changes made to this document are listed in the following table. For more information, please contact [dochelp@microsoft.com](mailto:dochelp@microsoft.com).

| Section | Description | Revision class |
| --- | --- | --- |
| [2.3.7.8](#Section_7e266b6d32e54201b6878ec40c24cd73) EMR\_EXTCREATEFONTINDIRECTW Record | 11303 : Clarified how the elw field is interpreted depending on its length. | Major |

# Index

A

[Applicability](#section_6ef9488d7050458b8fe7a65829615640) 20

[ArcDirection enumeration](#section_9886c62f17884118894a76d5a8733685) 28

[ArmStyle enumeration](#section_3ae0077a0c9d4e26bc2d773858bb0ac3) 29

B

[BackgroundMode enumeration](#section_2f57054c2b694303a6f85aacd401f759) 29

[BitFIX28\_4 Object](#section_075e694f901848ada22c81dd07c67c69) 49

[BitFIX28\_4 packet](#section_075e694f901848ada22c81dd07c67c69) 49

[BitmapRecordTypes packet](#section_2a57d4bdf3dc45df95fd447382b5262a) 75

[Byte ordering](#section_8c517124fe8244e6b0b6c64cdcbe07cb) 19

[Byte ordering example](#section_26465dd0ab564321b5c96d93bc179945) 217

C

[Change tracking](#section_ba9bf555d72a403bbab206329f16a496) 278

[ClippingRecordTypes packet](#section_0ca0d18e324e452f9a4126e1a82e3e03) 102

[ColorAdjustment enumeration](#section_c80a94d5415d4c4c83b812c0eb98b81f) 30

[ColorAdjustment Object](#section_71d7f4d0b6454de8a5588a2e0fc596cb) 49

[ColorAdjustment packet](#section_71d7f4d0b6454de8a5588a2e0fc596cb) 49

[ColorMatchToTarget enumeration](#section_b63c02c176444ef3a4963127621cb7b5) 30

[ColorSpace enumeration](#section_75da12f8f43748e886774b46e0045969) 30

[CommentRecordTypes packet](#section_05940d07e1124146ac0588fc6a1f70b9) 107

[Common data types and fields](#section_c16b82c9e35b4cd0bf074752c7599936) 21

[Contrast enumeration](#section_652d38bb5626462caf8ba5dcb828d814) 31

[ControlRecordTypes packet](#section_2caf0f4781a44079931dfca6fa499869) 115

D

[Data types and fields - common](#section_c16b82c9e35b4cd0bf074752c7599936) 21

[DesignVector Object](#section_cdf6681fbf454eec923e08c8be16aaf2) 51

[DesignVector packet](#section_cdf6681fbf454eec923e08c8be16aaf2) 51

Details

[common data types and fields](#section_c16b82c9e35b4cd0bf074752c7599936) 21

[DIBColors enumeration](#section_a5e722e3891a4a67be1aed5a48a7fda1) 31

[Drawing Record Types](#section_1f7da2e4df0441d48f7d180fa642cab4) 124

[DrawingRecordTypes packet](#section_1f7da2e4df0441d48f7d180fa642cab4) 124

E

[EMF metafile example](#section_2d7b37b4e17042f6bd0b3638d456549c) 218

[EMF Metafile Example example](#section_2d7b37b4e17042f6bd0b3638d456549c) 218

[EMF Metafile Playback example](#section_50ae16a09d1748018bf1e28e9533a889) 212

[EmfMetafileHeader packet](#section_80c0684b9a0e4bdb8d6d0e5a9e9673e7) 120

[EmfMetafileHeader Record](#section_80c0684b9a0e4bdb8d6d0e5a9e9673e7) 120

[EmfMetafileHeaderExtension1 packet](#section_120a68e1e1994768ba1bb6b0bd019989) 121

[EmfMetafileHeaderExtension1 Record](#section_120a68e1e1994768ba1bb6b0bd019989) 121

[EmfMetafileHeaderExtension2 packet](#section_3b393b9124db4daab5bac74434a1d67c) 122

[EmfMetafileHeaderExtension2 Record](#section_3b393b9124db4daab5bac74434a1d67c) 122

[EMR\_ALPHABLEND packet](#section_34e07d4faee64b63a4bb96996ad47669) 77

[EMR\_ALPHABLEND record](#section_34e07d4faee64b63a4bb96996ad47669) 77

[EMR\_ANGLEARC packet](#section_3b04a2e29ba5477fb79c5710635d04b9) 128

[EMR\_ANGLEARC Record](#section_3b04a2e29ba5477fb79c5710635d04b9) 128

[EMR\_ARC packet](#section_3e148f197e7543aa9259fda562f60315) 129

[EMR\_ARC Record](#section_3e148f197e7543aa9259fda562f60315) 129

[EMR\_ARCTO packet](#section_51b4a890bfd4496ebbd9dbde9ad49d1c) 130

[EMR\_ARCTO Record](#section_51b4a890bfd4496ebbd9dbde9ad49d1c) 130

EMR\_BITBLT example ([section 3.2.4](#section_ccd6a1e9f5f9452db7c7ed361149b7fa) 235, [section 3.2.6](#section_57892014242146d787d9380b972a5547) 237)

[EMR\_BITBLT packet](#section_347d1c44184747ec87627059e9e9b185) 82

[EMR\_BITBLT Record](#section_347d1c44184747ec87627059e9e9b185) 82

[EMR\_CHORD packet](#section_b442d2fbf92b4103aa7d44f864f719d5) 131

[EMR\_CHORD Record](#section_b442d2fbf92b4103aa7d44f864f719d5) 131

[EMR\_COLORCORRECTPALETTE packet](#section_99de1bfb3d77455b86797386903e1472) 179

[EMR\_COLORCORRECTPALETTE record](#section_99de1bfb3d77455b86797386903e1472) 179

[EMR\_COLORMATCHTOTARGETW packet](#section_b9e3e8fa1d074475b0f2d157eeca5417) 191

[EMR\_COLORMATCHTOTARGETW record](#section_b9e3e8fa1d074475b0f2d157eeca5417) 191

[EMR\_COMMENT packet](#section_e081b202429d4c34b21ca0ad501858a6) 108

[EMR\_COMMENT Record](#section_e081b202429d4c34b21ca0ad501858a6) 108

[EMR\_COMMENT\_BEGINGROUP packet](#section_9f0561a0adfe4a97a45299664787c1d2) 111

[EMR\_COMMENT\_BEGINGROUP Record](#section_9f0561a0adfe4a97a45299664787c1d2) 111

[EMR\_COMMENT\_EMFPLUS packet](#section_929b78e1b84844a59fac327cae5c2ae5) 109

[EMR\_COMMENT\_EMFPLUS Record](#section_929b78e1b84844a59fac327cae5c2ae5) 109

[EMR\_COMMENT\_EMFSPOOL packet](#section_a1a907e3cd444475a8cfdfad513a4243) 109

[EMR\_COMMENT\_EMFSPOOL Record](#section_a1a907e3cd444475a8cfdfad513a4243) 109

[EMR\_COMMENT\_ENDGROUP packet](#section_11829e5dd2594179b66bbc2c7ab6b985) 112

[EMR\_COMMENT\_ENDGROUP Record](#section_11829e5dd2594179b66bbc2c7ab6b985) 112

[EMR\_COMMENT\_MULTIFORMATS packet](#section_4bc4608eca82426dac0e5ded2ad3fc4e) 113

[EMR\_COMMENT\_MULTIFORMATS Record](#section_4bc4608eca82426dac0e5ded2ad3fc4e) 113

[EMR\_COMMENT\_PUBLIC packet](#section_39c91f22e4cc4e82a6050137caa8457c) 110

[EMR\_COMMENT\_PUBLIC Record Types](#section_39c91f22e4cc4e82a6050137caa8457c) 110

[EMR\_COMMENT\_WINDOWS\_METAFILE packet](#section_0dbabbd2c61d4534b526b9431c494da8) 114

[EMR\_COMMENT\_WINDOWS\_METAFILE Record](#section_0dbabbd2c61d4534b526b9431c494da8) 114

[EMR\_CREATEBRUSHINDIRECT example](#section_a0a54fae6d2a4dfa90c576578b084392) 233

[EMR\_CREATEBRUSHINDIRECT packet](#section_b9a8ef5d00894e42b317e6ebc0ff098f) 168

[EMR\_CREATEBRUSHINDIRECT Record](#section_b9a8ef5d00894e42b317e6ebc0ff098f) 168

[EMR\_CREATECOLORSPACE packet](#section_954af0ffa8d74d3480ed89f570bac016) 169

[EMR\_CREATECOLORSPACE record](#section_954af0ffa8d74d3480ed89f570bac016) 169

[EMR\_CREATECOLORSPACEW packet](#section_d1cd21477fb64d978f7ea289006b6064) 169

[EMR\_CREATECOLORSPACEW record](#section_d1cd21477fb64d978f7ea289006b6064) 169

[EMR\_CREATEDIBPATTERNBRUSHPT packet](#section_332116b4c6f94b18a7cc22c531b52afc) 170

[EMR\_CREATEDIBPATTERNBRUSHPT Record](#section_332116b4c6f94b18a7cc22c531b52afc) 170

[EMR\_CREATEMONOBRUSH packet](#section_49b4227731b04eb9a6af86d9be9b568f) 172

[EMR\_CREATEMONOBRUSH Record](#section_49b4227731b04eb9a6af86d9be9b568f) 172

[EMR\_CREATEPALETTE packet](#section_07e1492be4bb4394934f4eaee67ab8ff) 173

[EMR\_CREATEPALETTE Record](#section_07e1492be4bb4394934f4eaee67ab8ff) 173

[EMR\_CREATEPEN packet](#section_2374647fdf6748e386aa384715c28e71) 174

[EMR\_CREATEPEN Record](#section_2374647fdf6748e386aa384715c28e71) 174

[EMR\_DELETECOLORSPACE packet](#section_5d137387d79a4bc89a4d38291320e148) 180

[EMR\_DELETECOLORSPACE record](#section_5d137387d79a4bc89a4d38291320e148) 180

EMR\_DELETEOBJECT example ([section 3.2.15](#section_dd92ecb044d649e48ed4cc2179dac6fa) 263, [section 3.2.19](#section_7a4251dfa2a44ec6abb9a2fbcde3e0c1) 268, [section 3.2.20](#section_bb48d8904cc948a3888834e1bb71e32d) 268)

[EMR\_DELETEOBJECT packet](#section_6f0f12a3111a478b8251a9505168f9a9) 180

[EMR\_DELETEOBJECT Record](#section_6f0f12a3111a478b8251a9505168f9a9) 180

[EMR\_DRAWESCAPE packet](#section_86405bb1cc2c42bc9e0f28874f208d98) 164

[EMR\_DRAWESCAPE Record](#section_86405bb1cc2c42bc9e0f28874f208d98) 164

[EMR\_ELLIPSE packet](#section_d9462a32d18840e7bf7205614ec4ff18) 132

[EMR\_ELLIPSE Record](#section_d9462a32d18840e7bf7205614ec4ff18) 132

[EMR\_EOF example](#section_e955bcd5d9494b89b88023aca87a23a9) 269

[EMR\_EOF packet](#section_3f47fde00e6b40c187f3f4129af03aa1) 116

[EMR\_EOF Record](#section_3f47fde00e6b40c187f3f4129af03aa1) 116

[EMR\_EXCLUDECLIPRECT packet](#section_a521411eb8774199abb6b4514b3574f8) 104

[EMR\_EXCLUDECLIPRECT Record](#section_a521411eb8774199abb6b4514b3574f8) 104

EMR\_EXTCREATEFONTINDIRECTW example ([section 3.2.8](#section_99afd5f1cf274c55874533daae20875d) 251, [section 3.2.11](#section_d95d812067e34d20a3c09e620d860a7e) 256, [section 3.2.13](#section_d6fb56d91e754a05a970b177ab1ea5ec) 260, [section 3.2.16](#section_a3c8a88677f943088cb7d967e340416a) 264)

[EMR\_EXTCREATEFONTINDIRECTW packet](#section_7e266b6d32e54201b6878ec40c24cd73) 175

[EMR\_EXTCREATEFONTINDIRECTW Record](#section_7e266b6d32e54201b6878ec40c24cd73) 175

[EMR\_EXTCREATEPEN packet](#section_d7f51e054024497cad4a8aeca9d34256) 176

[EMR\_EXTCREATEPEN Record](#section_d7f51e054024497cad4a8aeca9d34256) 176

[EMR\_EXTESCAPE packet](#section_a31ef1e754ba41aeb79de6fd102dc6b1) 165

[EMR\_EXTESCAPE Record](#section_a31ef1e754ba41aeb79de6fd102dc6b1) 165

[EMR\_EXTFLOODFILL packet](#section_4f1b52ba325a4da8bd5b987c4a572e5f) 132

[EMR\_EXTFLOODFILL Record](#section_4f1b52ba325a4da8bd5b987c4a572e5f) 132

[EMR\_EXTSELECTCLIPRGN packet](#section_c6b9f4e627f64a4da383c2daf5da11d9) 105

[EMR\_EXTSELECTCLIPRGN Record](#section_c6b9f4e627f64a4da383c2daf5da11d9) 105

[EMR\_EXTTEXTOUTA packet](#section_6b582a713c294fc6a0f41f8a313739a1) 133

[EMR\_EXTTEXTOUTA record](#section_6b582a713c294fc6a0f41f8a313739a1) 133

[EMR\_EXTTEXTOUTW example](#section_624eb0057ca749a9a17037be301e0424) 255

[EMR\_EXTTEXTOUTW packet](#section_a59a79ac328e492da34de02727af6edf) 134

[EMR\_EXTTEXTOUTW Record](#section_a59a79ac328e492da34de02727af6edf) 134

[EMR\_FILLPATH packet](#section_7cc0afcb06934810a3cbb69c871ce473) 135

[EMR\_FILLPATH Record](#section_7cc0afcb06934810a3cbb69c871ce473) 135

[EMR\_FILLRGN packet](#section_a1bb0f88bb224956b45a7005546561cc) 135

[EMR\_FILLRGN Record](#section_a1bb0f88bb224956b45a7005546561cc) 135

[EMR\_FORCEUFIMAPPING packet](#section_e9a05924e704403783126bef2e19f6bd) 192

[EMR\_FORCEUFIMAPPING Record](#section_e9a05924e704403783126bef2e19f6bd) 192

[EMR\_FRAMERGN packet](#section_c370fc169045467ba1257239428df6b3) 136

[EMR\_FRAMERGN Record](#section_c370fc169045467ba1257239428df6b3) 136

[EMR\_GLSBOUNDEDRECORD packet](#section_cb5036dac79d4aaeb1c0a48361447a0b) 185

[EMR\_GLSBOUNDEDRECORD record](#section_cb5036dac79d4aaeb1c0a48361447a0b) 185

[EMR\_GLSRECORD packet](#section_4f2e25120c5d468bbb81484cb1d4aa89) 185

[EMR\_GLSRECORD record](#section_4f2e25120c5d468bbb81484cb1d4aa89) 185

[EMR\_GRADIENTFILL packet](#section_1a3849c8be6c4d30b5d3f43b4c70ca0d) 137

[EMR\_GRADIENTFILL record](#section_1a3849c8be6c4d30b5d3f43b4c70ca0d) 137

[EMR\_HEADER example](#section_ae7e7437cfe5485e84eac74b51b000be) 231

[EMR\_HEADER packet](#section_de081cd7351f4cc2830bd03fb55e89ab) 117

[EMR\_HEADER Record Types](#section_de081cd7351f4cc2830bd03fb55e89ab) 117

[EMR\_INTERSECTCLIPRECT packet](#section_13cd0c98d4e94ca7a79d58055bf45c79) 105

[EMR\_INTERSECTCLIPRECT Record](#section_13cd0c98d4e94ca7a79d58055bf45c79) 105

[EMR\_INVERTRGN packet](#section_5cbc44dead4047b6a0f86179b9eef738) 192

[EMR\_INVERTRGN Record](#section_5cbc44dead4047b6a0f86179b9eef738) 192

[EMR\_LINETO packet](#section_9b3eccf94a554a9cb0df3c495e7b9a8c) 139

[EMR\_LINETO Record](#section_9b3eccf94a554a9cb0df3c495e7b9a8c) 139

[EMR\_MASKBLT packet](#section_e6f715c0d0344eb3952ef8ee66adb9ed) 84

[EMR\_MASKBLT Record](#section_e6f715c0d0344eb3952ef8ee66adb9ed) 84

[EMR\_MODIFYWORLDTRANSFORM packet](#section_c70b85e58c31418fa7b8349e417e0f76) 210

[EMR\_MODIFYWORLDTRANSFORM Record](#section_c70b85e58c31418fa7b8349e417e0f76) 210

[EMR\_MOVETOEX packet](#section_8ae57e5a22e3435298ca09628f47c693) 193

[EMR\_MOVETOEX Record](#section_8ae57e5a22e3435298ca09628f47c693) 193

[EMR\_NAMEDESCAPE packet](#section_125759e0c8394dc1b5fbe014e7e1159e) 165

[EMR\_NAMEDESCAPE Record](#section_125759e0c8394dc1b5fbe014e7e1159e) 165

[EMR\_OFFSETCLIPRGN packet](#section_8bf2b60d3b194bd1b2d7c89b027ad808) 106

[EMR\_OFFSETCLIPRGN Record](#section_8bf2b60d3b194bd1b2d7c89b027ad808) 106

[EMR\_PAINTRGN packet](#section_68c457dc6df34609b6f9e28e5ca4d9c1) 140

[EMR\_PAINTRGN Record](#section_68c457dc6df34609b6f9e28e5ca4d9c1) 140

[EMR\_PIE packet](#section_c326f6b2ae7e447fb8dc02c703f054c3) 140

[EMR\_PIE Record](#section_c326f6b2ae7e447fb8dc02c703f054c3) 140

[EMR\_PIXELFORMAT packet](#section_ba9aa909eb054d2fa51b10cb6df68a54) 194

[EMR\_PIXELFORMAT record](#section_ba9aa909eb054d2fa51b10cb6df68a54) 194

[EMR\_PLGBLT packet](#section_b6c8b39d42c74221ab7093f1e09b644e) 88

[EMR\_PLGBLT Record](#section_b6c8b39d42c74221ab7093f1e09b644e) 88

[EMR\_POLYBEZIER packet](#section_4e53793a95af49d4ae1f4c407eda9440) 141

[EMR\_POLYBEZIER Record](#section_4e53793a95af49d4ae1f4c407eda9440) 141

[EMR\_POLYBEZIER16 packet](#section_3329ee2617ed43718f51e3985d764d77) 142

[EMR\_POLYBEZIER16 Record](#section_3329ee2617ed43718f51e3985d764d77) 142

[EMR\_POLYBEZIERTO packet](#section_28431e45a87441dc864d8f4f69e8e831) 143

[EMR\_POLYBEZIERTO Record](#section_28431e45a87441dc864d8f4f69e8e831) 143

[EMR\_POLYBEZIERTO16 packet](#section_d27cc96684a44715a6de45aa6cd03292) 144

[EMR\_POLYBEZIERTO16 Record](#section_d27cc96684a44715a6de45aa6cd03292) 144

[EMR\_POLYDRAW packet](#section_95794fafeed84d5bb6ffc6765a58166e) 145

[EMR\_POLYDRAW Record](#section_95794fafeed84d5bb6ffc6765a58166e) 145

[EMR\_POLYDRAW16 packet](#section_57f0cfe721394199b6ad61fb61a1d4ec) 146

[EMR\_POLYDRAW16 Record](#section_57f0cfe721394199b6ad61fb61a1d4ec) 146

[EMR\_POLYGON packet](#section_eb91678158b64e92b60668071aa65733) 147

[EMR\_POLYGON Record](#section_eb91678158b64e92b60668071aa65733) 147

[EMR\_POLYGON16 packet](#section_ac6d65ac9d39439382ebb58fa0c0f635) 148

[EMR\_POLYGON16 Record](#section_ac6d65ac9d39439382ebb58fa0c0f635) 148

[EMR\_POLYLINE packet](#section_9ce6c9bb1a1348a59aa2d95b334b5358) 149

[EMR\_POLYLINE Record](#section_9ce6c9bb1a1348a59aa2d95b334b5358) 149

[EMR\_POLYLINE16 packet](#section_ca3a6590c02343548d6de3d40d7eb5b9) 150

[EMR\_POLYLINE16 Record](#section_ca3a6590c02343548d6de3d40d7eb5b9) 150

[EMR\_POLYLINETO packet](#section_a2d8b73883514a9c9f3aa6a8481c4c6f) 150

[EMR\_POLYLINETO Record](#section_a2d8b73883514a9c9f3aa6a8481c4c6f) 150

[EMR\_POLYLINETO16 packet](#section_0d5a471025f943bebb083e38a4db1249) 151

[EMR\_POLYLINETO16 Record](#section_0d5a471025f943bebb083e38a4db1249) 151

[EMR\_POLYPOLYGON packet](#section_c60ff127271142d385d4502ca2e4caef) 152

[EMR\_POLYPOLYGON Record](#section_c60ff127271142d385d4502ca2e4caef) 152

[EMR\_POLYPOLYGON16 packet](#section_96ad447f25d642728d1851b395e669c6) 153

[EMR\_POLYPOLYGON16 Record](#section_96ad447f25d642728d1851b395e669c6) 153

[EMR\_POLYPOLYLINE packet](#section_e23f75fd3af649f38c86ef121ff61701) 154

[EMR\_POLYPOLYLINE Record](#section_e23f75fd3af649f38c86ef121ff61701) 154

[EMR\_POLYPOLYLINE16 packet](#section_a8b4cdfd50394a56b7cb9458f819f23f) 156

[EMR\_POLYPOLYLINE16 Record](#section_a8b4cdfd50394a56b7cb9458f819f23f) 156

[EMR\_POLYTEXTOUTA packet](#section_d8b7ac2176b84f9aa7cc05f9d2d5627e) 156

[EMR\_POLYTEXTOUTA record](#section_d8b7ac2176b84f9aa7cc05f9d2d5627e) 156

[EMR\_POLYTEXTOUTW packet](#section_381015eb29a7477493ada210697f5972) 158

[EMR\_POLYTEXTOUTW record](#section_381015eb29a7477493ada210697f5972) 158

[EMR\_RECTANGLE packet](#section_3c4712380a02499290a2bfd2afd98f2a) 159

[EMR\_RECTANGLE Record](#section_3c4712380a02499290a2bfd2afd98f2a) 159

[EMR\_RESIZEPALETTE packet](#section_ec78933296b54bb89d7c8b5c8c0da8b9) 181

[EMR\_RESIZEPALETTE Record](#section_ec78933296b54bb89d7c8b5c8c0da8b9) 181

[EMR\_RESTOREDC packet](#section_efa2f1804c7e4a7bb4efa1d06a0ce89f) 194

[EMR\_RESTOREDC Record](#section_efa2f1804c7e4a7bb4efa1d06a0ce89f) 194

[EMR\_ROUNDRECT packet](#section_2e4f55a15f694a818329423eeedb3812) 159

[EMR\_ROUNDRECT Record](#section_2e4f55a15f694a818329423eeedb3812) 159

[EMR\_SCALEVIEWPORTEXTEX packet](#section_469e353c42094919a5aa9331b60765ed) 195

[EMR\_SCALEVIEWPORTEXTEX Record](#section_469e353c42094919a5aa9331b60765ed) 195

[EMR\_SCALEWINDOWEXTEX packet](#section_01698f29bbcb41009bad0fd52c33b0da) 195

[EMR\_SCALEWINDOWEXTEX Record](#section_01698f29bbcb41009bad0fd52c33b0da) 195

[EMR\_SELECTCLIPPATH packet](#section_4a26bcf866074a098ec3a8768eadc8e8) 106

[EMR\_SELECTCLIPPATH Record](#section_4a26bcf866074a098ec3a8768eadc8e8) 106

EMR\_SELECTOBJECT example ([section 3.2.3](#section_27a971dd9f6c457e9d99a9548bb86f27) 234, [section 3.2.5](#section_76c92930196d4a13a63f4731a8823d6a) 237, [section 3.2.9](#section_7da1d80b58004d5f88ae272ca6004950) 254, [section 3.2.12](#section_08ad22157aee4bd18a1a60ee3624455d) 259, [section 3.2.14](#section_048e4b0204a24a89a56dbfe35a6508c2) 263, [section 3.2.17](#section_1cf96f7a8eed40b8be557143833089c2) 267, [section 3.2.18](#section_577db3263d3a4320b01f311a3f40327d) 267, [section 3.2.21](#section_6155fa84449c4559ba37f0f1c19d665e) 268)

[EMR\_SELECTOBJECT packet](#section_145b063d5f9641feb7ae1e615b2bc2bf) 181

[EMR\_SELECTOBJECT Record](#section_145b063d5f9641feb7ae1e615b2bc2bf) 181

[EMR\_SELECTPALETTE packet](#section_e6a4ce2a209d43dfb7635d8e54c21a10) 182

[EMR\_SELECTPALETTE Record](#section_e6a4ce2a209d43dfb7635d8e54c21a10) 182

[EMR\_SETARCDIRECTION packet](#section_83a955e613b84720a521542e3dd91cba) 196

[EMR\_SETARCDIRECTION Record](#section_83a955e613b84720a521542e3dd91cba) 196

[EMR\_SETBKCOLOR packet](#section_24f4095262d440b0ba40aad8fb4fc421) 197

[EMR\_SETBKCOLOR Record](#section_24f4095262d440b0ba40aad8fb4fc421) 197

[EMR\_SETBKMODE example](#section_db2224f4dd2843f1801a28144e4eef1b) 251

[EMR\_SETBKMODE packet](#section_7c584a42b6dd47b8a0b6f529133292f4) 197

[EMR\_SETBKMODE Record](#section_7c584a42b6dd47b8a0b6f529133292f4) 197

[EMR\_SETBRUSHORGEX packet](#section_b02ab7ee4c884bf29ce99100636d86be) 198

[EMR\_SETBRUSHORGEX Record](#section_b02ab7ee4c884bf29ce99100636d86be) 198

[EMR\_SETCOLORADJUSTMENT packet](#section_81b3cd7ced6742aaab9847b99d697231) 198

[EMR\_SETCOLORADJUSTMENT Record](#section_81b3cd7ced6742aaab9847b99d697231) 198

[EMR\_SETCOLORSPACE packet](#section_2a84d7a5f8c14dd2ae79a029a25ad601) 182

[EMR\_SETCOLORSPACE record](#section_2a84d7a5f8c14dd2ae79a029a25ad601) 182

[EMR\_SETDIBITSTODEVICE packet](#section_e8816cc635d243e68d88d69cd342372e) 92

[EMR\_SETDIBITSTODEVICE Record](#section_e8816cc635d243e68d88d69cd342372e) 92

[EMR\_SETICMMODE packet](#section_a7c9d4834a2440588dc1b5e0809ea758) 199

[EMR\_SETICMMODE record](#section_a7c9d4834a2440588dc1b5e0809ea758) 199

[EMR\_SETICMPROFILEA packet](#section_cf1bc41cbdc54645890fc03e05d4d26d) 199

[EMR\_SETICMPROFILEA record](#section_cf1bc41cbdc54645890fc03e05d4d26d) 199

[EMR\_SETICMPROFILEW packet](#section_1d376b73e0084440b6b95770d4afdee5) 200

[EMR\_SETICMPROFILEW record](#section_1d376b73e0084440b6b95770d4afdee5) 200

[EMR\_SETLAYOUT packet](#section_dcb69f9d6f69466aaf1e6f505d86cc67) 201

[EMR\_SETLAYOUT record](#section_dcb69f9d6f69466aaf1e6f505d86cc67) 201

[EMR\_SETLINKEDUFIS packet](#section_3bb62157624f443fa88cef5e696740dd) 202

[EMR\_SETLINKEDUFIS Record](#section_3bb62157624f443fa88cef5e696740dd) 202

[EMR\_SETMAPMODE packet](#section_aa4ad35dfa424a4f959a8b41304e1b05) 202

[EMR\_SETMAPMODE Record](#section_aa4ad35dfa424a4f959a8b41304e1b05) 202

[EMR\_SETMAPPERFLAGS packet](#section_6647abf7bd7043ef87661a6e06ed2f03) 203

[EMR\_SETMAPPERFLAGS record](#section_6647abf7bd7043ef87661a6e06ed2f03) 203

[EMR\_SETMITERLIMIT packet](#section_2637d0d092dd48e8be5e6400b4d1fc5e) 203

[EMR\_SETMITERLIMIT Record](#section_2637d0d092dd48e8be5e6400b4d1fc5e) 203

[EMR\_SETPALETTEENTRIES packet](#section_883482963c9a488fbbf719c897535372) 183

[EMR\_SETPALETTEENTRIES Record](#section_883482963c9a488fbbf719c897535372) 183

[EMR\_SETPIXELV packet](#section_21d44fb9b3c74ce7a4027601f34d0cda) 160

[EMR\_SETPIXELV Record](#section_21d44fb9b3c74ce7a4027601f34d0cda) 160

[EMR\_SETPOLYFILLMODE packet](#section_e8799c80fe7445c1bf6d3eb78edd4ed2) 204

[EMR\_SETPOLYFILLMODE Record](#section_e8799c80fe7445c1bf6d3eb78edd4ed2) 204

[EMR\_SETROP2 packet](#section_cedee7e04e6647b6ae1d816e1e319786) 204

[EMR\_SETROP2 Record](#section_cedee7e04e6647b6ae1d816e1e319786) 204

[EMR\_SETSTRETCHBLTMODE packet](#section_ff00ab986e2c4ecab66404ab2956b6a5) 205

[EMR\_SETSTRETCHBLTMODE Record](#section_ff00ab986e2c4ecab66404ab2956b6a5) 205

[EMR\_SETTEXTALIGN packet](#section_a49bb565a58341da8d4a8a0314b3e397) 205

[EMR\_SETTEXTALIGN Record](#section_a49bb565a58341da8d4a8a0314b3e397) 205

[EMR\_SETTEXTCOLOR record](#section_1ae5c616cf8e4f5da7c3bfc1f70ece28) 206

[EMR\_SETTEXTJUSTIFICATION packet](#section_edbc2be01da045d69c05677cbcaa2d47) 206

[EMR\_SETVIEWPORTEXTEX packet](#section_a71486a39ebd43e3819f5318a58b70d5) 207

[EMR\_SETVIEWPORTEXTEX Record](#section_a71486a39ebd43e3819f5318a58b70d5) 207

[EMR\_SETVIEWPORTORGEX packet](#section_b89489473f8d4adda4b8e572e8f90506) 207

[EMR\_SETVIEWPORTORGEX Record](#section_b89489473f8d4adda4b8e572e8f90506) 207

[EMR\_SETWINDOWEXTEX packet](#section_1952148360b549b8b58c4b5d1b6cf839) 208

[EMR\_SETWINDOWEXTEX Record](#section_1952148360b549b8b58c4b5d1b6cf839) 208

[EMR\_SETWINDOWORGEX packet](#section_952803a93d5c4c83a258558bc1123bdb) 208

[EMR\_SETWINDOWORGEX Record](#section_952803a93d5c4c83a258558bc1123bdb) 208

[EMR\_SETWORLDTRANSFORM packet](#section_985724c04db148f0b34667288b3288cb) 211

[EMR\_SETWORLDTRANSFORM Record](#section_985724c04db148f0b34667288b3288cb) 211

[EMR\_SMALLTEXTOUT packet](#section_20eee81d0bd442d1a624860adfe62358) 161

[EMR\_SMALLTEXTOUT Record](#section_20eee81d0bd442d1a624860adfe62358) 161

[EMR\_STRETCHBLT packet](#section_0dd551f380f7485289c29ddba803a192) 94

[EMR\_STRETCHBLT Record](#section_0dd551f380f7485289c29ddba803a192) 94

[EMR\_STRETCHDIBITS packet](#section_89c0d8080dea413fbe402e9e51fa36ac) 97

[EMR\_STRETCHDIBITS Record](#section_89c0d8080dea413fbe402e9e51fa36ac) 97

[EMR\_STROKEANDFILLPATH packet](#section_2794792a38b44d19adec28fc2a6273b2) 162

[EMR\_STROKEANDFILLPATH Record](#section_2794792a38b44d19adec28fc2a6273b2) 162

[EMR\_STROKEPATH packet](#section_c8b86df6746445b0bca46ffab9174961) 163

[EMR\_STROKEPATH Record](#section_c8b86df6746445b0bca46ffab9174961) 163

[EMR\_TRANSPARENTBLT packet](#section_aa2160512dc04317b343525431cfa103) 100

[EMR\_TRANSPARENTBLT record](#section_aa2160512dc04317b343525431cfa103) 100

[EmrComment enumeration](#section_cd57866bd2ef49f09dd1d928ddc02885) 32

[EmrFormat Object](#section_1bd63563ce674528bf285a8852ff0270) 51

[EmrFormat packet](#section_1bd63563ce674528bf285a8852ff0270) 51

[EmrText Object](#section_dd585d0a5d7c4034963a1141af836972) 52

[EmrText packet](#section_dd585d0a5d7c4034963a1141af836972) 52

[Enumerations](#section_3ce820d7fb1f426b94d61c22bad3a176) 21

[EpsData Object](#section_a624b402f7aa4cb6b289612c9775f37e) 54

[EpsData packet](#section_a624b402f7aa4cb6b289612c9775f37e) 54

[EscapeRecordTypes packet](#section_5370873a5fe64fdd8c5abade4ee54a1f) 163

Examples

[byte ordering example](#section_26465dd0ab564321b5c96d93bc179945) 217

[EMF Metafile Example](#section_2d7b37b4e17042f6bd0b3638d456549c) 218

[EMF Metafile Playback](#section_50ae16a09d1748018bf1e28e9533a889) 212

EMR\_BITBLT example ([section 3.2.4](#section_ccd6a1e9f5f9452db7c7ed361149b7fa) 235, [section 3.2.6](#section_57892014242146d787d9380b972a5547) 237)

[EMR\_CREATEBRUSHINDIRECT example](#section_a0a54fae6d2a4dfa90c576578b084392) 233

EMR\_DELETEOBJECT example ([section 3.2.15](#section_dd92ecb044d649e48ed4cc2179dac6fa) 263, [section 3.2.19](#section_7a4251dfa2a44ec6abb9a2fbcde3e0c1) 268, [section 3.2.20](#section_bb48d8904cc948a3888834e1bb71e32d) 268)

[EMR\_EOF example](#section_e955bcd5d9494b89b88023aca87a23a9) 269

EMR\_EXTCREATEFONTINDIRECTW example ([section 3.2.8](#section_99afd5f1cf274c55874533daae20875d) 251, [section 3.2.11](#section_d95d812067e34d20a3c09e620d860a7e) 256, [section 3.2.13](#section_d6fb56d91e754a05a970b177ab1ea5ec) 260, [section 3.2.16](#section_a3c8a88677f943088cb7d967e340416a) 264)

[EMR\_EXTTEXTOUTW example](#section_624eb0057ca749a9a17037be301e0424) 255

[EMR\_HEADER example](#section_ae7e7437cfe5485e84eac74b51b000be) 231

EMR\_SELECTOBJECT example ([section 3.2.3](#section_27a971dd9f6c457e9d99a9548bb86f27) 234, [section 3.2.5](#section_76c92930196d4a13a63f4731a8823d6a) 237, [section 3.2.9](#section_7da1d80b58004d5f88ae272ca6004950) 254, [section 3.2.12](#section_08ad22157aee4bd18a1a60ee3624455d) 259, [section 3.2.14](#section_048e4b0204a24a89a56dbfe35a6508c2) 263, [section 3.2.17](#section_1cf96f7a8eed40b8be557143833089c2) 267, [section 3.2.18](#section_577db3263d3a4320b01f311a3f40327d) 267, [section 3.2.21](#section_6155fa84449c4559ba37f0f1c19d665e) 268)

[EMR\_SETBKMODE example](#section_db2224f4dd2843f1801a28144e4eef1b) 251

[metafile design examples](#section_50ae16a09d1748018bf1e28e9533a889) 212

[ExtTextOutOptions enumeration](#section_e7ffcc5340d148738edac5c5ee104aa5) 32

F

[FamilyType enumeration](#section_cfb9abd0a0a446dfb782eb856e19a5d3) 33

[Fields - vendor-extensible](#section_23e918a03cc74afe9197094930db3d16) 20

[FloodFill enumeration](#section_e9e1009705f540d190ac79ccbe0b5fa9) 34

[FormatSignature enumeration](#section_b55cbd24612d47fe81c91da6a682246f) 34

G

[Glossary](#section_6de331ec81f64ab28982673a232e0a5c) 10

[GradientFill enumeration](#section_699e75a028af4d6fb7aadc76e1b0001a) 34

[GradientRectangle Object](#section_93ba94f784fc4dd58c943d476aa7588b) 55

[GradientRectangle packet](#section_93ba94f784fc4dd58c943d476aa7588b) 55

[GradientTriangle Object](#section_9aa3346cdc29448d9dcf6a4f6b77d229) 55

[GradientTriangle packet](#section_9aa3346cdc29448d9dcf6a4f6b77d229) 55

[GraphicsMode enumeration](#section_fe60a5a523044b639949e11bfdbebae9) 35

H

[HatchStyle enumeration](#section_db3f9395eb7540a7ba9f0f836658712d) 36

[Header Object](#section_e4a35c41e8e343f9bc07a18e99bb866d) 56

[Header packet](#section_e4a35c41e8e343f9bc07a18e99bb866d) 56

[HeaderExtension1 Object](#section_00cc8ab4ea2e4bb195691201af47a0c8) 57

[HeaderExtension1 packet](#section_00cc8ab4ea2e4bb195691201af47a0c8) 57

[HeaderExtension2 Object](#section_9e96e5cfe94949aebaa83fffd948e588) 58

[HeaderExtension2 packet](#section_9e96e5cfe94949aebaa83fffd948e588) 58

I

[ICMMode enumeration](#section_4935fb3cc6464d16a8f0f86ee0a9d096) 36

[Illuminant enumeration](#section_7cd7679e4a1445d19cacbf7f80be2c10) 36

[Implementer - security considerations](#section_9677eccd8a8d4480959e46b8dc747f4c) 270

[Informative references](#section_6cd585e907854a389c701ce9e990ee92) 17

[Introduction](#section_e889f1bfc12349aca8e29eebbba13754) 10

L

[Letterform enumeration](#section_ee9127c4596f4744a59f44d9c3c5760f) 37

[Localization](#section_7d8af617d0344c2b99908f126ecad50d) 20

[LogBrushEx Object](#section_6d790f8d51654746bac33443915c2071) 58

[LogBrushEx packet](#section_6d790f8d51654746bac33443915c2071) 58

[LogFont Object](#section_1cd7cb7a1a7c48d2b4a4c218f9d12100) 59

[LogFont packet](#section_1cd7cb7a1a7c48d2b4a4c218f9d12100) 59

[LogFontEx Object](#section_71516231003445f2af78c463b98db4af) 61

[LogFontEx packet](#section_71516231003445f2af78c463b98db4af) 61

[LogFontExDv Object](#section_595fcaaebc3741489afe859e81ca7f76) 62

[LogFontExDv packet](#section_595fcaaebc3741489afe859e81ca7f76) 62

[LogFontPanose Object](#section_c68ffff81c324398ac812cb6f0cd7a7c) 62

[LogFontPanose packet](#section_c68ffff81c324398ac812cb6f0cd7a7c) 62

[LogPalette Object](#section_758f047d765e424c9204a833b7b4e527) 64

[LogPalette packet](#section_758f047d765e424c9204a833b7b4e527) 64

[LogPaletteEntry Object](#section_c1f7b285be164112a4e60b2fd4c1d148) 64

[LogPaletteEntry packet](#section_c1f7b285be164112a4e60b2fd4c1d148) 64

[LogPen Object](#section_93ce3f4537ac4affb6e82f6db054c4c4) 65

[LogPen packet](#section_93ce3f4537ac4affb6e82f6db054c4c4) 65

[LogPenEx Object](#section_5b67b3eeea004f809b732959804381be) 65

[LogPenEx packet](#section_5b67b3eeea004f809b732959804381be) 65

M

[MapMode enumeration](#section_109d7955767e4955a0e2f8b5e14f757a) 38

[Metafile design examples](#section_50ae16a09d1748018bf1e28e9533a889) 212

[Metafile structure](#section_daaf94470c47446eb72eac6bd7a2e8f1) 17

[MetafileVersion enumeration](#section_710a4199428649149f331ec1f30022c3) 39

[MidLine enumeration](#section_27eb84dc85b14a9dbf9a15808b2c61d8) 39

[ModifyWorldTransformMode enumeration](#section_e6bb2996195f473f80c69dc1afe474f9) 40

[MR\_SETTEXTCOLOR packet](#section_1ae5c616cf8e4f5da7c3bfc1f70ece28) 206

N

[Normative references](#section_bce0eb192dc94f119109e8dc000c7784) 16

O

[ObjectCreationRecordTypes packet](#section_7d914464d5464393bddb6d1e747fca18) 166

[ObjectManipulationRecordTypes packet](#section_2f1c3268b43b4c28988b41aa49354998) 177

[Objects](#section_8380c4b942324f6d93a8e7a4426f38a5) 49

[OpenGLRecordTypes packet](#section_4a6c2fc239af43b48254449f5f1c5c05) 184

[Overview (synopsis)](#section_26cabe550e604309876c93e161f76747) 17

P

[Panose Object](#section_4f1d246221334247ae1b2192cd91adb6) 67

[Panose packet](#section_4f1d246221334247ae1b2192cd91adb6) 67

[Path Bracket Record Types](#section_b930b98930ec4954a8891dc60ce0b689) 186

[PathBracketRecordTypes packet](#section_b930b98930ec4954a8891dc60ce0b689) 186

[PenStyle enumeration](#section_b2e629158401429fbb6db2506a4c2fe8) 40

[PixelFormatDescriptor Object](#section_1db036d62da84b92b4f8e9cab8cc93b7) 68

[PixelFormatDescriptor packet](#section_1db036d62da84b92b4f8e9cab8cc93b7) 68

[Point enumeration](#section_01b63e3f19d6437c8fc62758884bc08d) 41

[Point28\_4 Object](#section_c66bda3b65ae4b3f8c8f62ad7f94f730) 71

[Point28\_4 packet](#section_c66bda3b65ae4b3f8c8f62ad7f94f730) 71

[PolygonFillMode enumeration](#section_879a186908ad497783e809be265e2c7c) 42

[Product behavior](#section_f5148c7c30c54d95bf1bb449c1f8f53d) 271

[Proportion enumeration](#section_d59152c4deb049d88eeb0407167698fe) 42

R

[RecordType enumeration](#section_1eec80ba799b4784a9ac91597d590ae1) 21

[References](#section_0862960b50b0447ba4ae841946cea5ae) 16

[informative](#section_6cd585e907854a389c701ce9e990ee92) 17

[normative](#section_bce0eb192dc94f119109e8dc000c7784) 16

[RegionData Object](#section_e66601f29b5c46198476ddb7b087551b) 71

[RegionData packet](#section_e66601f29b5c46198476ddb7b087551b) 71

[RegionDataHeader Object](#section_5ca68a15181145b6a51b5e40d1055ccc) 72

[RegionDataHeader packet](#section_5ca68a15181145b6a51b5e40d1055ccc) 72

[RegionMode enumeration](#section_b7f99f50dd2f45289624f74140368019) 43

[Relationship to other protocols](#section_fd7a5d7883a642159cba597079527f11) 19

[Relationship to protocols and other structures](#section_fd7a5d7883a642159cba597079527f11) 19

S

[Security](#section_9677eccd8a8d4480959e46b8dc747f4c) 270

[Security - implementer considerations](#section_9677eccd8a8d4480959e46b8dc747f4c) 270

[SerifType enumeration](#section_750aae5cb2574e15acfac1be27b8c408) 43

[StateRecordTypes packet](#section_8f8a4df2f8d74d39afc094e19f524652) 187

[StockObject enumeration](#section_d6dffd25861542f8aed1309f1fe54ab2) 44

[StretchMode enumeration](#section_e72f20aa2e2d453f9e95eaef50b76e19) 47

[StrokeVariation enumeration](#section_08b1f3afc0c14ebf823a757ade94df22) 47

Structures

[Drawing Record Types](#section_1f7da2e4df0441d48f7d180fa642cab4) 124

[enumerations](#section_3ce820d7fb1f426b94d61c22bad3a176) 21

[objects](#section_8380c4b942324f6d93a8e7a4426f38a5) 49

[overview](#section_c16b82c9e35b4cd0bf074752c7599936) 21

[Path Bracket Record Types](#section_b930b98930ec4954a8891dc60ce0b689) 186

T

[Tracking changes](#section_ba9bf555d72a403bbab206329f16a496) 278

[TransformRecordTypes packet](#section_a0442ad12d6e43e7ac8ce252e492f97e) 209

[TriVertex Object](#section_33578509834946b68f8f107c3f70bace) 72

[TriVertex packet](#section_33578509834946b68f8f107c3f70bace) 72

U

[UniversalFontId Object](#section_34502dab22b44855b9b288f00a4a0c52) 73

[UniversalFontId packet](#section_34502dab22b44855b9b288f00a4a0c52) 73

V

[Vendor-extensible fields](#section_23e918a03cc74afe9197094930db3d16) 20

[Versioning](#section_7d8af617d0344c2b99908f126ecad50d) 20

W

[Weight enumeration](#section_46caba8a83e5450d8c4dd3999be38b93) 48

X

[XForm Object](#section_e84107e9bc2b4a1492345d173adc1b59) 74

[XForm packet](#section_e84107e9bc2b4a1492345d173adc1b59) 74

[XHeight enumeration](#section_47e9daaf0fff4ea09b4fc30329c79b64) 48