Network Working Group
Internet-Draft
Intended Status: Informational
Expires: April 20, 2015

S. Leonard
Penango, Inc.

October 17, 2014

The Windows Metafile and Enhanced Metafile Media Types
draft-seantek-image-wmf-emf-00

Abstract

This document registers the image/wmf and image/emf media types for use with Windows Metafile and Enhanced Metafile formats. Originally designed for Microsoft Windows 3.0, these image files are intended to be portable between applications and devices, and may contain both vector and raster graphics.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

1. Introduction

Long before the invention of Scalable Vector Graphics, Microsoft Corporation recognized the value of recording images in a format that its applications and operating systems could easily render irrespective of the output device. With the release of Windows 3.0, Microsoft released its Windows Metafile (WMF) format, which can contain vector and raster graphics in one package. As a binary format that needed to work on 16-bit machines, WMF is comprised of a sequence of record structures. Each record contains drawing commands, object definitions, and configuration settings. When a metafile is processed, the image can be rendered on a display, output to a printer or plotter, stored in memory, or saved to some persistent storage. Reflecting the relationship to the Windows Graphics Device Interface (GDI) API, WMF metafiles are "played" to a playback device context in the same manner that PostScript content is treated as an executable program that results in the output of the original document.

As Microsoft’s first 32-bit operating system, Windows NT 3.1 introduced an overhaul to the Windows API ("Win32") and the in-memory formats upon which those APIs relied. The Enhanced Metafile (EMF) format was created at this time, using 32-bit values instead of WMF’s 16-bit values. In Windows XP, Microsoft extended EMF with "EMF+", adding support for Windows GDI+.

Many implementations of WMF and EMF were created because of Windows’ commercial success in the 1990s. A large body of free and commercially available clip art and other artwork exists in this format. Furthermore, WMF content appears non-normatively in certain standards (e.g., [OOXML]); the usage is common enough that an implementer would almost certainly need to support it for basic interoperability.

Microsoft publicly documented the WMF format as early as the 1992 Windows 3.1 SDK. Since 2007 Microsoft has released the format specifications [MS-WMF] and [MS-EMF] under its Open Specification Promise [MS-OSP].

The key word "SHOULD" in this document is to be interpreted as described in [RFC2119].

2. Windows Metafile Media Type Registration Application

Type name: image

Subtype name: wmf

Required parameters: None.

Optional parameters:

DEFAULT_CHARSET: The character set intended when the CharacterSet Enumeration (see [MS-WMF]) specifies DEFAULT_CHARSET. The value of this parameter is a charset name defined in accordance to the procedures laid out in [RFC2978]. When this parameter is not specified, DEFAULT_CHARSET has the following meaning in [MS-WMF]: "a character set based on the current system locale; for example, when the system locale is United States English, the default character set is ANSI_CHARSET" (which is Windows-1252, more-or-less). I.e., when not specified, the default character set is system-dependent. As this optional parameter is novel, content producers embedding text SHOULD use EMF instead of WMF (or if absolutely necessary, SHOULD embed EMF within WMF).

Encoding considerations: Binary.

Security considerations:

The Windows Metafile format’s security history is punctuated in 2005-2006 with the disclosure of the Metafile Image Code Execution vulnerability, codenamed MICE. MICE won the 2007 Pwnie Award for "Mass 0wnage" and "Breaking the Internet" [PWNIES07]. The official Microsoft security bulletin [MICE] describes that the flaw occurs because Windows Metafiles can set the SETABORTPROC value of the MetafileEscapes enumeration (accessible via the META_ESCAPE record), allowing for arbitrary code execution.

Windows Metafiles can contain Enhanced Metafiles using the META_ESCAPE_ENHANCED_METAFILE record; thus, the security considerations of EMF apply to WMF.

Windows Metafiles are historically very buggy. As the original intent was to replicate Windows GDI calls, flaws in GDI, or in a display or printer driver implementing the back-end to GDI, could be exploitable. WMF implementations not backed by Windows GDI have different risks: namely, while a malicious WMF author may not consider the non-Windows GDI implementation as a primary target, WMF has many "corner case" records for which an implementation’s processing may not have received the same level of scrutiny as the Windows implementation. "Fuzzing" the implementation is appropriate.

Interoperability considerations:

Windows Metafile is the original 16-bit metafile format; it was

released in 1990 at what some computer historians might consider the "zenith" of the desktop publishing revolution. Accordingly, there is a large body of free and commercially available clip art that is still in use, either independently or embedded in productivity documents (word processing documents, desktop publishing documents, slideshows and presentations, and spreadsheets and workbooks). For example, references to WMF content appear (non-normatively) in Office Open XML [OOXML]. To say that support for this format is necessary for interoperability would not be an understatement.

Accommodations for comments or arbitrary data storage in Windows Metafiles are virtually non-existent. However, Windows Metafiles can contain Enhanced Metafiles using the META_ESCAPE_ENHANCED_METAFILE record; an implementation SHOULD be able to handle both types. Windows Metafiles can store and output text strings (see META_TEXTOUT and META_EXTTEXTOUT records), but the encodings of the strings may be ambiguous. Unicode encodings are not possible without the DEFAULT_CHARSET parameter defined in this registration.

The previously unregistered type "image/x-wmf" is also in wide use. Accordingly, it is registered as a deprecated alias. See Appendix A and Section 4.2.9 of [RFC6838].

Published specification: [MS-WMF].

Applications that use this media type:

Office productivity applications; clip art applications; desktop publishing applications; some Web browsers (e.g., Internet Explorer).

Fragment identifier considerations: None.

Additional information:

Deprecated alias names for this type: image/x-wmf
Magic number(s): D7 CD C6 9A (little-endian DWORD 0x9AC6CDD7)
File extension(s): .wmf
Macintosh file type code(s):
????. A uniform type identifier (UTI) of "????" is RECOMMENDED.

Person & email address to contact for further information:

Sean Leonard <dev+ietf@seantek.com>

Restrictions on usage: None.

Author/Change controller: Sean Leonard <dev+ietf@seantek.com>

Intended usage: COMMON

Provisional registration? No

3. Enhanced Metafile Media Type Registration Application

Type name: image

Subtype name: emf

Required parameters: None.

Optional parameters: None.

Encoding considerations: Binary.

Security considerations:

Enhanced Metafiles are not afflicted with [MICE]. There has been no public disclosure of vulnerabilities specific to EMF or EMF+ to date. Nonetheless:

Enhanced Metafiles can contain Encapsulated PostScript (EPS) data; thus the security considerations of PostScript processing may also apply to EMF.

As the original intent was to replicate Windows GDI calls, flaws in GDI, or in a display or printer driver implementing the back-end to GDI, could be exploitable with maliciously crafted EMF content. EMF implementations not backed by Windows GDI have different risks: namely, while a malicious EMF author may not consider the non-Windows GDI implementation as a primary target, EMF has many "corner case" records for which an implementation’s processing may not have received the same level of scrutiny as the Windows implementation. "Fuzzing" the implementation is appropriate. It is also possible that EMF+ data is "safe" while EMF data contains an exploit (or vice-versa); the EMF+-aware implementation (such as an application designed for GDI+ on Windows XP or above) would skip the "unsafe" data while another implementation would fall prey to the exploit.

Interoperability considerations:

Enhanced Metafile is the 32-bit metafile format; it was released in 1992 along with Windows NT 3.1. There is a large body of free and commercially available clip art that is still in use, either

independently or embedded in productivity documents (word processing documents, desktop publishing documents, slideshows and presentations, and spreadsheets and workbooks). To say that support for this format is necessary for interoperability would not be an understatement.

Enhanced Metafiles have extensive accommodations for comments and arbitrary data storage. Enhanced Metafiles can store and output text strings. Mercifully, the encodings of these strings are well-defined. Record examples include EMR_EXTTEXTOUTA (US-ASCII), EMR_EXTTEXTOUTW (UTF16-LE), EMR_POLYTEXTOUTA (US-ASCII), EMR_POLYTEXTOUTW (UTF16-LE), and EMR_SMALLTEXTOUT (UTF16-LE or the low-order 8 bits of UTF16-LE--effectively ISO-8859-1--depending on ETO_SMALL_CHARS).

Enhanced Metafiles can contain Encapsulated PostScript (EPS) data in the EpsData object [MS-EMF]. The FormatSignature EPS_SIGNATURE (0x46535045, in little-endian) is used instead of ENHMETA_SIGNAUTRE (0x464D4520, in little-endian) in such a case.

Windows XP introduced the GDI+ API, along with EMF+ [MS-EMF+]. EMF+ is actually an embedded format in which GDI+ commands are stored as EMF comment records (EMR_COMMENT_EMFPLUS record type). Content containing EMF+ data can be identified as "EMF+ Only" (only EMF+; the EMF records are not sufficient to reconstitute the drawing) or "EMF+ Dual" (both EMF records alone or EMF+ records alone, when played back, are sufficient to reconstitute the drawing) [MS-EMF+]. Support for EMF+ records may not be as extensive as support for the original EMF records.

The previously unregistered type "image/x-emf" is also in wide use. Accordingly, it is registered as a deprecated alias. See Appendix A and Section 4.2.9 of [RFC6838].

Published specification: [MS-EMF] and [MS-EMF+].

Applications that use this media type:

Office productivity applications; clip art applications; desktop publishing applications; some Web browsers (e.g., Internet Explorer).

Fragment identifier considerations: None.

Additional information:

Deprecated alias names for this type: image/x-emf
Magic number(s): 01 00 00 00 (little-endian DWORD 0x00000001),

                    corresponding to the EMR_HEADER Type field.
                    The next field (EMR_HEADER Size) should be
                    at least 88 (little-endian DWORD 0x00000050).
      File extension(s): .emf
                         (for both EMF and EMF+ content)
      Macintosh file type code(s):
        ????. A uniform type identifier (UTI) of "????" is RECOMMENDED.

   Person & email address to contact for further information:

     Sean Leonard <dev+ietf@seantek.com>

   Restrictions on usage: None.

   Author/Change controller: Sean Leonard <dev+ietf@seantek.com>

   Intended usage: COMMON

   Provisional registration? No

4. IANA Considerations

   IANA is asked to register the media types image/wmf and image/emf in the Standards tree using the applications provided in Sections 2 and 3 of this document.

5. Security Considerations

   See the image/wmf and image/emf registration templates for their respective security considerations.

6. References

6.1. Normative References

   [MS-WMF]   Microsoft Corporation, "Windows Metafile Format", [MS-WMF], v20140502 (Rev 11.1), May 2014, <http://msdn.microsoft.com/library/cc250370>.

   [MS-EMF]   Microsoft Corporation, "Enhanced Metafile Format", [MS-EMF], v20140502 (Rev 10.0), May 2014, <http://msdn.microsoft.com/library/cc230514>.

   [MS-EMF+] Microsoft Corporation, "Enhanced Metafile Format Plus Extensions", [MS-EMFPLUS], v20140502 (Rev 13.0), May 2014, <http://msdn.microsoft.com/library/cc230724>.

   [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate

Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2978] Freed, N. and J. Postel, "IANA Charset Registration Procedures", BCP 19, RFC 2978, October 2000.

[RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, January 2013.

6.2. Informative References

[MICE] Microsoft Corporation, "Vulnerability in Graphics Rendering Engine Could Allow Remote Code Execution (912919)", MS06-001, V1.0, January 2006, <https://technet.microsoft.com/library/security/ms06-001>.

[MS-OSP] Microsoft Corporation, "Open Specification Promise", February 2007, <http://www.microsoft.com/interop/osp/default.mspx>.

[OOXML] Ecma International, "Office Open XML File Formats", Standard ECMA-376, Fourth Edition, ISO/IEC 29500, December 2012, <http://www.ecma-international.org/publications/standards/Ecma-376.htm>.

[PWNIES07] Pwnie Awards LLC, "Pwnie Awards 2007", 2007, <http://pwnies.com/archive/2007/winners/>.

Author’s Address

Sean Leonard
Penango, Inc.
5900 Wilshire Boulevard
21st Floor
Los Angeles, CA 90036
USA

EMail: dev+ietf@seantek.com
URI: http://www.penango.com/