Independent Submission
Request for Comments: 7903
Category: Informational
ISSN: 2070-1721

S. Leonard
Penango, Inc.
September 2016

Windows Image Media Types

Abstract

This document registers media types for certain image formats promulgated in Microsoft Windows, namely image/wmf, image/x-wmf, image/emf, image/x-emf, and image/bmp for use with Windows Metafile, Enhanced Metafile, and Windows Bitmap formats. Originally designed for Microsoft Windows 2.0 and 3.0, these image files are intended to be portable between applications and devices, and they may contain both vector and raster graphics.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction 2
1.1. Windows Metafiles 2
1.2. Windows Bitmaps 3
2. Windows Metafile Media Type Registration Application 4
3. Enhanced Metafile Media Type Registration Application 6
4. Windows Bitmap Media Type Registration Application 9
5. IANA Considerations 11
6. Security Considerations 11
7. References 11
7.1. Normative References 11
7.2. Informative References 11
Author's Address 12

1. Introduction

1.1. Windows Metafiles

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 data is "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], [MS-EMF], and [MS-EMF+] under its Open Specification Promise [MS-OSP].

1.2. Windows Bitmaps

Long before the invention of Portable Network Graphics (PNG), Microsoft Corporation and IBM Corporation needed to record images in a format that their applications and operating systems could easily render on low-end machines (Intel 80286). The resulting "BMP" format contains a single raster graphic with basic header fields that can be easily mapped (or "blitted") to locations in memory. As computing moved from 16-bit to 32-bit, BMP evolved to contain 32-bit structures. As the ’90s wore on, the venerable BMP got boosts with support for additional color spaces, color profiles, and compression formats. The same basic format can be used to convey 2-bit black-and-white bitmaps with a 1-bit alpha mask from the ’80s, and full-color Ultra HD images on leading-edge displays. BMP is a building block of other formats, including Windows Metafiles, Windows Icons, and Windows Cursors.

Many implementations of BMP were created because of Windows’ commercial success in the 1990s. Usage of the format for interchange has declined since the advent of PNG (for lossless raster graphics) and JPEG (for lossy raster graphics); however, a large body of free and commercially available BMP artwork exists. Since Windows Icons are a building block of "favicon.ico" web technology, an implementer would almost certainly need to support this format for basic interoperability.

Microsoft publicly documented the BMP format as early as the 1992 Windows 3.1 SDK (in the Windows Metafile documentation). Since 2007 Microsoft has released the format specification [MS-WMF], which includes most components of the Windows Bitmap format, under its Open Specification Promise [MS-OSP]. See Section 2.2.2.9 of [MS-WMF] (DeviceIndependentBitmap Object). BMP data begins with a BITMAPFILEHEADER and is followed by one of the bitmap headers (BITMAPINFOHEADER, BITMAPV4HEADER, or BITMAPV5HEADER), optional color table data, bitmap data, and optional profile data, in that order [BMPSTOR].

Implementers need to be aware of the [MICE] vulnerability, and to guard against it. Some details are included in the completed registration template.

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 the WMF specification) specifies DEFAULT_CHARSET. The value of this parameter is a charset from the IANA "Character Sets" registry <http://www.iana.org/assignments/character-sets>. When this parameter is not specified, DEFAULT_CHARSET has the following meaning in the WMF specification: "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). That is, when not specified, the default character set is system dependent. This optional parameter is new to this registration and may not enjoy widespread support for some time. Therefore, EMF instead of WMF (or if necessary under the circumstances, embedded EMF within WMF) is a more sensible choice when text is present.

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 ("MICE") vulnerability. MICE won the 2007 Pwnie Award for "Mass 0wnage" and "Breaking the Internet". The official Microsoft security bulletin 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, i.e., "active content".

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 backend 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.

As a "basic" image format, the image/wmf media type does not employ executable content and provides no facilities for privacy or integrity.

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, presentations, spreadsheets, and workbooks). For example, references to WMF content appear (non-normatively) in Office Open XML. 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, so an implementation that handles Windows Metafiles is also expected to handle enhanced metafile content. 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.

Published specification:

WMF: Microsoft Corporation, "[MS-WMF]: Windows Metafile Format", v20160714 (Rev 13.1), July 2016, <http://msdn.microsoft.com/library/cc250370>.

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):
None. A uniform type identifier (UTI) of "com.microsoft.wmf" is suggested.

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 the Metafile Image Code Execution ("MICE") vulnerability. There has been no public disclosure of vulnerabilities specific to EMF or EMF+ to date. Neither EMF nor EMF+ are designed to contain "active content". 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 backend 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.

As a "basic" image format, the image/emf media type does not employ executable content and provides no facilities for privacy or integrity.

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, presentations, 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. 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+. 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). 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.

Published specification:

EMF: Microsoft Corporation, "[MS-EMF]: Enhanced Metafile Format", v20160714 (Rev 12.0), July 2016, <http://msdn.microsoft.com/library/cc230514>.

EMF+: Microsoft Corporation, "[MS-EMFPLUS]: Enhanced Metafile Format Plus Extensions", v20160714 (Rev 14.1), July 2016, <http://msdn.microsoft.com/library/cc230724>.

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):
None. A uniform type identifier (UTI) of "com.microsoft.emf" is suggested.

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. Windows Bitmap Media Type Registration Application

Type name: image

Subtype name: bmp

Required parameters: None.

Optional parameters: None.

Encoding considerations: Binary.

Security considerations:

Bitmaps have a mostly unremarkable security history.

Because BMP data can encapsulate JPEG or PNG data (BI_JPEG, BI_PNG values of the Compression enumeration in Section 2.1.1.7 of the WMF specification), the security considerations of JPEG and PNG processing may also apply to BMP.

As a "basic" image format, the image/bmp media type does not employ executable content and provides no facilities for privacy or integrity.

Interoperability considerations:

Uncompressed Windows Bitmaps can be rather large. If there is a need to compress an image, modern applications should consider emitting JPEG or PNG data instead of embedding them in BMP payloads.

Published specification:

WMF: Microsoft Corporation, "[MS-WMF]: Windows Metafile Format", v20160714 (Rev 13.1), July 2016, <http://msdn.microsoft.com/library/cc250370>.

BMP: Microsoft Corporation, "Bitmap Storage", MSDN ID dd183391, <http://msdn.microsoft.com/library/dd183391>.

Applications that use this media type:

Office productivity applications; clip art applications; desktop publishing applications; web browsers; graphics processing applications.

Fragment identifier considerations: None.

Additional information:

Magic number(s):
42 4D ("BM"), meaning "bitmap". The next field (BITMAPFILEHEADER bfSize) is a little-endian DWORD indicating the size of the bitmap content in bytes.

File extension(s): .bmp, .dib

Macintosh file type code(s):
"BMP ", "BMPf", or "BMPp". Apple has promulgated a uniform type identifier (UTI) of "com.microsoft.bmp".

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

5. IANA Considerations

IANA has registered the media types image/wmf, image/x-wmf, image/emf, image/x-emf, and image/bmp in the standards tree using the applications provided in Sections 2, 3, and 4 of this document.

6. Security Considerations

See the registration templates for their respective security considerations.

The Metafile Image Code Execution (MICE) vulnerability won the 2007 Pwnie Award for "Mass 0wnage" and "Breaking the Internet" [PWNIES07].

7. References

7.1. Normative References

[BMPSTOR] Microsoft Corporation, "Bitmap Storage", MSDN ID dd183391, <http://msdn.microsoft.com/library/dd183391>.

[MS-EMF] Microsoft Corporation, "[MS-EMF]: Enhanced Metafile Format", v20160714 (Rev 12.0), July 2016, <http://msdn.microsoft.com/library/cc230514>.

[MS-EMF+] Microsoft Corporation, "[MS-EMFPLUS]: Enhanced Metafile Format Plus Extensions", v20160714 (Rev 14.1), July 2016, <http://msdn.microsoft.com/library/cc230724>.

[MS-WMF] Microsoft Corporation, "[MS-WMF]: Windows Metafile Format", v20160714 (Rev 13.1), July 2016, <http://msdn.microsoft.com/library/cc250370>.

7.2. 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, <https://msdn.microsoft.com/en-us/openspecifications/dn646765>.

[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, "Winners of 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
United States of America

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