[MS-OLEDS]:
Object Linking and Embedding (OLE) Data Structures

The OLE File Format Data Structures allow data from one application to be stored in the document of another application.

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.

- **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 may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.

- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.

- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft's Open Specification Promise (available here: http://www.microsoft.com/interop/osp) or the Community Promise (available here: http://www.microsoft.com/interop/cp/default.mspx). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.

- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.

- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events 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 specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications do 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 are intended for use in conjunction with publicly available standard

specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

Revision Summary

Date	Revision History	Revision Class	Comments
04/08/2008	0.1		Initial availability.
06/20/2008	0.1.1	Editorial	Revised and edited the technical content.
07/25/2008	0.1.2	Editorial	Revised and edited the technical content.
08/29/2008	1.0	Major	Updated and revised the technical content.
10/24/2008	1.0.1	Editorial	Revised and edited the technical content.
12/05/2008	1.1	Minor	Updated the technical content.
01/16/2009	1.1.1	Editorial	Revised and edited the technical content.
02/27/2009	1.1.2	Editorial	Revised and edited the technical content.
04/10/2009	1.1.3	Editorial	Revised and edited the technical content.
05/22/2009	1.1.4	Editorial	Revised and edited the technical content.
07/02/2009	1.1.5	Editorial	Revised and edited the technical content.
08/14/2009	2.0	Major	Updated and revised the technical content.
09/25/2009	3.0	Major	Updated and revised the technical content.
11/06/2009	3.0.1	Editorial	Revised and edited the technical content.
12/18/2009	3.0.2	Editorial	Revised and edited the technical content.
01/29/2010	4.0	Major	Updated and revised the technical content.
03/12/2010	4.0.1	Editorial	Revised and edited the technical content.
04/23/2010	5.0	Major	Updated and revised the technical content.
06/04/2010	5.0.1	Editorial	Revised and edited the technical content.
07/16/2010	5.0.1	No change	No changes to the meaning, language, or formatting of the technical content.
08/27/2010	5.0.1	No change	No changes to the meaning, language, or formatting of the technical content.
10/08/2010	5.0.1	No change	No changes to the meaning, language, or formatting of the technical content.
11/19/2010	5.0.1	No change	No changes to the meaning, language, or formatting of

Date	Revision History	Revision Class	Comments
			the technical content.
01/07/2011	5.0.1	No change	No changes to the meaning, language, or formatting of the technical content.
02/11/2011	5.0.1	No change	No changes to the meaning, language, or formatting of the technical content.

Contents

1 Introduction

This document specifies the Object Linking and Embedding (OLE) Data Structures:

- The Object Linking and Embedding (OLE) Data Structures allow data from one application to be stored in the document of another application. The first application is called the **creating application** and the second application is called the **container application**. The data itself is called an **embedded object**.

- The OLE File Format Data Structures also allow a document from one application to reference data that resides in a different application. The first application is called the container application and the second application is called the creating application. The data being referenced is called a **linked object**.

For example, a user can embed a spreadsheet (which is data that belongs to the spreadsheet application) in a word-processing document. When the word-processing application displays the document to the user, it can establish that the spreadsheet data belongs to the spreadsheet application and the word-processing application can interact with the spreadsheet application to display the spreadsheet data to the user.

1.1 Glossary

The following terms are defined in [MS-GLOS]:

American National Standards Institute (ANSI) character set
class identifier (CLSID)
drive letter
little-endian
path
Unicode
Unicode string
Universal Naming Convention (UNC)

The following terms are specific to this document:

clipboard: A program provided by the operating system that enables local data transfer between applications by using the cut, copy, and paste operations.

container application: An application that creates documents that store reference data from other applications.

container document: A document that stores or references data from more than one application.

creating application: An application whose data is stored in or referenced by documents from other applications.

embedded object: Application data that is stored in documents from other applications.

linked object: Application data that is referenced by documents from other applications.

native data: The data that constitutes the state of an **embedded object**. The only entity that can create and process the data is the **creating application**.

presentation data: Data that is required to display the linked or **embedded object** within the **container application**.

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as described in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

1.2.1 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. We will assist you in finding the relevant information. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

[MS-CFB] Microsoft Corporation, "Compound File Binary File Format", June 2008.

[MS-DTYP] Microsoft Corporation, "Windows Data Types", January 2007.

[MS-EMF] Microsoft Corporation, "Enhanced Metafile Format", July 2007.

[MS-ERREF] Microsoft Corporation, "Windows Error Codes", January 2007.

[MS-RPRN] Microsoft Corporation, "Print System Remote Protocol Specification", June 2007.

[MS-WMF] Microsoft Corporation, "Windows Metafile Format", June 2007.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.ietf.org/rfc/rfc2119.txt

1.2.2 Informative References

[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary", March 2007.

[MSDN-ADVF] Microsoft Corporation, "ADVF", http://msdn.microsoft.com/en-us/library/ms693742(VS.85).aspx

[MSDN-COM] Microsoft Corporation, "Component Object Model", http://msdn.microsoft.com/en-us/library/aa286559.aspx

[MSDN-CP] Microsoft Corporation, "Code Page Identifiers", http://msdn.microsoft.com/en-us/library/dd317756(VS.85).aspx

If you have any trouble finding [MSDN-CP], please check here.

[MSDN-CREATEDC] Microsoft Corporation, "CreateDC Function", http://msdn.microsoft.com/en-us/library/dd183490(v=VS.85).aspx

[MSDN-DVAPSECT] Microsoft Corporation, "DVAPSECT", http://msdn.microsoft.com/en-us/library/ms690318(VS.85).aspx

[MSDN-FRMATCLNDX] Microsoft Corporation, "FORMATETCLINDEX", http://msdn.microsoft.com/en-us/library/ms682177(vs.85).aspx

[MSDN-IPersistStream] Microsoft Corporation, "IPersistStream", http://msdn.microsoft.com/en-us/library/ms690091.aspx

[MSDN-IStream] Microsoft Corporation, "IStream Interface", http://msdn.microsoft.com/en-us/library/aa380034.aspx

[MSDN-OLEUPDATE] Microsoft Corporation, "OLEUPDATE", http://msdn.microsoft.com/en-us/library/ms679715.aspx

[MSDN-ProgID] Microsoft Corporation, "<ProgID> Key", http://msdn.microsoft.com/en-us/library/ms690196.aspx

1.3 Overview

The Object Linking and Embedding (OLE) Data Structures enable applications to create documents that contain linked or embedded objects. The file format of an embedded object is different from that of a linked object because the embedded object is required to contain its **native data** as well as data about its creating application. The linked object, however, is required only to contain a reference to its data. Both embedded and linked objects are required to contain **presentation data**.

The following sections describe the logical layout of the file formats of embedded and linked objects.

1.3.1 Embedded Objects

The logical layout of a **container document** that contains an embedded object is shown in the following figure.

Container Application Data	Creating Application Data	Embedded Object Native Data	Embedded Object Presentation Data

Container Document
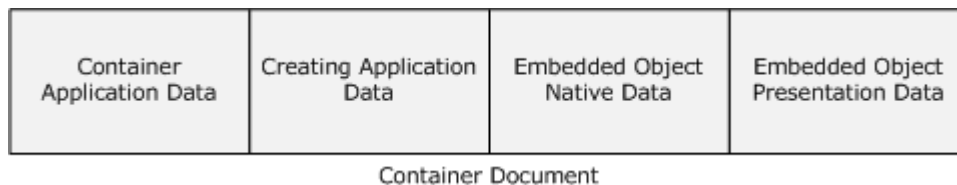
Figure 1: Embedded object container document

The document contains data that is specific to the container application, an identifier of the creating application, the native data of the embedded object, and the presentation data of the embedded object. The native data allows the container application to pass the object back to its creating application for possible changes.

The container application data is private to the container application and is not specified in this document. The rest of the data that make up the embedded object file format are specified in this document as follows:

Creating Application Identifier

- When using the OLE1.0 Format (section 1.3.3), the creating application identifier is specified in the **ClassName** field of the ObjectHeader structure (section 2.2.4).

- When using the OLE2.0 Format (section 1.3.3), the creating application identifier is specified as follows:

  - For embedded objects, the creating application identifier is specified by the object class globally unique identifier (GUID) contained in the **class identifier (CLSID)** field of the Compound File Directory Entry ([MS-CFB] section 2.6.1) of the OLE Compound File Storage

object ([MS-CFB] section 1.3) that contains the OLESTREAM structure (section 2.3.3) that specifies the embedded object.

- For linked objects, the creating application identifier is specified by the object class GUID contained in the **Clsid** field of the OLESTREAM structure (section 2.3.3) that specifies the linked object.

Embedded Object Native Data

- When using the OLE1.0 Format (section 1.3.3), the embedded object native data is specified in the **NativeData** field of EmbeddedObject structure (section 2.2.5).

- When using the OLE2.0 Format (section 1.3.3), the embedded object native data is specified in the following two ways, which can be used interchangeably:

  - An OLE Compound File Stream object that uses the name "\1Ole10Native" is created within the OLE Compound File Storage object that corresponds to the linked object or embedded object. The stream object is used to contain the native data, as specified in the **NativeData** field of the OLENativeStream structure (section 2.3.6).

  - OLE Compound File Stream objects are created by the creating application within the OLE Compound File Storage object that corresponds to the linked object or embedded object. The stream objects are used to contain the native data. Such stream objects are private to the creating application and are not specified in this document.

Embedded Object Presentation Data

- When using the OLE1.0 Format (section 1.3.3), the embedded object presentation data is specified in the **Presentation** field of EmbeddedObject structure (section 2.2.5).

- When using the OLE2.0 Format (section 1.3.3), the embedded object presentation data is specified in OLE Compound File Stream objects with names that contain the prefix "\2OlePres" within the OLE Compound File Storage object. The names correspond to the linked object or embedded object. Each of these streams contains an instance of the OLEPresentationStream structure (section 2.3.4).

1.3.2 Linked Objects

The logical layout of a container document that contains a linked object is shown in the following figure:
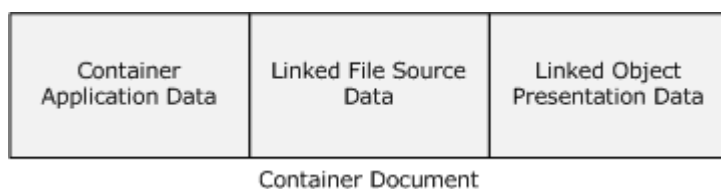
Figure 2: Linked object container document

The document contains data that is specific to the container application, data that describes the source file that contains the linked object's native data, and the presentation data for the linked object. The document does not contain the native data of the linked object.

The container application data is private to the container application and is not specified in this document. The rest of the data that make up the linked object format are specified in this document as follows:

Linked File Source Data

- When using the OLE1.0 Format (section 1.3.3), the Linked File Source Data is specified in the **TopicName** and the **NetworkName** fields of the LinkedObject structure (section 2.2.6).

- When using the OLE2.0 Format (section 1.3.3), the Linked File Source Data is specified by the **MonikerStream** field of the OLEStream structure (section 2.3.3).

Linked Object Presentation Data

- When using the OLE1.0 Format (section 1.3.3), the linked object presentation data is specified in the **Presentation** field of the LinkedObject structure (section 2.2.6).

- When using the OLE2.0 Format (section 1.3.3), the linked object presentation data is specified in the OLEPresentationStream structure (section 2.3.4).

1.3.3 OLE1.0 and OLE2.0 Formats

There are two different formats for representing linked objects and embedded objects within container documents. The first format is the OLE1.0 Format. This format predates the OLE Compound File technology (as specified in [MS-CFB]). When using the OLE1.0 Format, the linked object and embedded object data is laid out as a sequence of bytes within the container document. The data structures in section 2.2 specify the format of the data within the byte sequence.

The second format is the OLE2.0 Format. This format uses the OLE Compound File technology (as specified in [MS-CFB]). When using the OLE2.0 Format, the container application creates an OLE Compound File Storage object ([MS-CFB] section 1.3) for each linked object or embedded object. The linked object or embedded object data is contained in this storage in the form of OLE Compound File Stream objects ([MS-CFB] section 1.3). The data structures in section 2.3 specify the format of the data contained in the stream objects.

It is required that an application differentiate in advance whether it is processing a file that uses the OLE1.0 Format or the OLE2.0 Format. This information is local to the application and is not specified in this document.

It is strongly advised that implementations of this specification use the OLE2.0 Format when creating container documents. The OLE1.0 Format is specified to allow only for backward-compatible implementations.

1.3.4 Miscellaneous OLE2.0 Streams

In addition to the OLE2.0 streams mentioned in sections 1.3.1 and 1.3.2, a container document may also include another OLE2.0 stream:

CompObjStream (section 2.3.7).

1.3.5 Clipboard Formats

The users of container applications and creating applications may use the system **clipboard** to transfer data between applications. All data copied to a system clipboard must conform to a format specification, known as a Clipboard Format. Each Clipboard Format is identified by a unique numeric format ID.

1.3.5.1 Standard Clipboard Formats

This specification uses a number of Clipboard Formats called the Standard Clipboard Formats. These are specified in section 2.1.1. A standard Clipboard Format is identified by a numeric ID as specified in section 2.1.1.

1.4 Relationship to Protocols and Other Structures

The OLE2.0 Format uses the OLE Compound File Streams (as specified in [MS-CFB]) to store linked object and embedded object data in container documents.

1.5 Applicability Statement

The OLE File Format specification is appropriate when creating a single container document that contains data from multiple applications. It is applicable when one of the applications is the primary owner of the document and that application will always be asked to process the document first. It is not applicable if there are no applications with primary ownership of the document.

1.6 Versioning and Localization

There is only one version of the OLE File Format Data Structure. This data structure does not define locale-specific processes or data.

This document specifies two types of OLE File Formats: the OLE1.0 Format and the OLE2.0 Format. These two formats are independent and are not differentiated using a versioning mechanism.

1.7 Vendor-Extensible Fields

The Object Linking and Embedding Data Structures use **HRESULT** error codes. These **values** are taken from the Windows error number space as specified in [MS-ERREF]. Vendors SHOULD<1> reuse those values with their indicated meaning. Choosing any other value runs the risk of a collision in the future.

1.7.1 Registered Clipboard Formats

Vendors may create their own Clipboard Formats, called Registered Clipboard Formats. A Registered Clipboard Format is identified by a string that contains the name of the Clipboard Format (as specified in section 2.1.1). Vendors can define their own Registered Clipboard Formats provided that the string that identifies the Registered Clipboard Format is prefixed with the "OleExternal" string literal.

2 Structures

Field types in packet diagrams are defined by the packet diagram and the field descriptions. All of the fields in packet diagrams use **little-endian** byte ordering unless otherwise stated. There is no alignment padding between fields unless otherwise specified.

This protocol uses the following types specified in [MS-DTYP]:

Type	Reference
HRESULT	[MS-DTYP] section 2.2.18
LONG	[MS-DTYP] section 2.2.27
FILETIME	[MS-DTYP] section 2.3.1

Data structures in this document make use of **ANSI** characters. This structure specification does not mandate the use of any particular ANSI code page.<2>

2.1 Common Data Types

This section describes the structures that are common to both the OLE1.0 Format and the OLE2.0 Format.

2.1.1 Clipboard Formats

Standard Clipboard Formats and Registered Clipboard Formats (see sections 1.3.5.1 and 1.7.1 for more details) are used to identify presentation data formats.

A standard clipboard format identifier is of type unsigned long.

A registered clipboard format is identified by a LengthPrefixedAnsiString (section 2.1.4) or a LengthPrefixedUnicodeString (section 2.1.5).

The Object Linking and Embedding (OLE) Data Structures: Structure Specification defines the following standard clipboard format values to be used to identify presentation data formats:

Name	Value	Presentation Data Format
CF_BITMAP	0x00000002	Bitmap16 Object structure (as specified in [MS-WMF] section 2.2.2.1)
CF_METAFILEPICT	0x00000003	Windows metafile (as specified in [MS-WMF] section 1.3.1)
CF_DIB	0x00000008	DeviceIndependentBitmap Object structure (as specified in [MS-WMF] section 2.2.2.9)
CF_ENHMETAFILE	0x0000000E	Enhanced Metafile (as specified in [MS-EMF] section 1.3.1)

In addition, an application or higher level protocol MAY supply registered clipboard formats (section 1.7.1) to identify custom presentation data formats.

2.1.2 CLSID (Packet)

The packet version of the CLSID structure represents a class identifier (CLSID) in a serialized manner.

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
Data1																															
Data2																Data3															
Data4																															
...																															
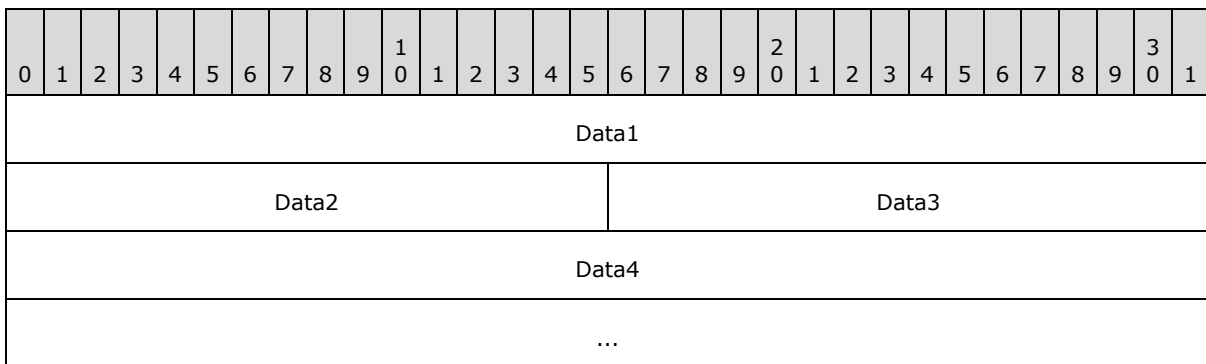
Data1 (4 bytes): This MUST be identical in meaning to the **Data1** field specified in **[MS-DTYP]** section 2.3.2.

Data2 (2 bytes): This MUST be identical in meaning to the **Data2** field specified in **[MS-DTYP]** section 2.3.2.

Data3 (2 bytes): This MUST be identical in meaning to the **Data3** field specified in **[MS-DTYP]** section 2.3.2.

Data4 (8 bytes): This MUST be identical in meaning to the **Data4** field specified in **[MS-DTYP]** section 2.3.2.

2.1.3 FILETIME (Packet)

The FILETIME (Packet) structure represents a **FILETIME** as specified in [MS-DTYP] section 2.3.1.

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
dwLowDateTime																															
dwHighDateTime																															
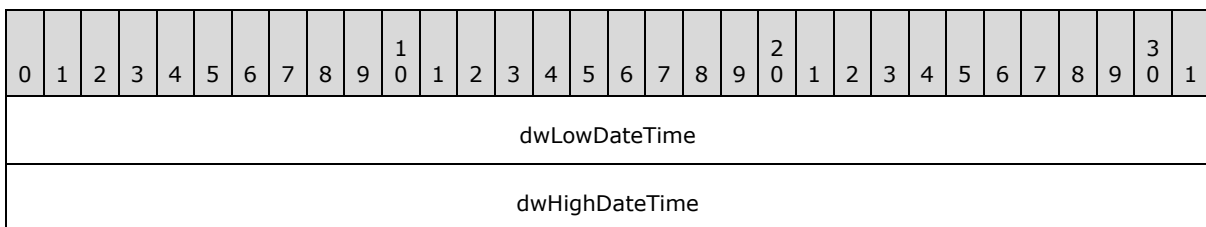
dwLowDateTime (4 bytes): This MUST be identical in meaning to the **dwLowDateTime** field specified in [MS-DTYP] section 2.3.1.

dwHighDateTime (4 bytes): This MUST be identical in meaning to the **dwHighDateTime** field specified in [MS-DTYP] section 2.3.1.

2.1.4 LengthPrefixedAnsiString

This structure specifies a null-terminated American National Standards Institute (ANSI) character set string.
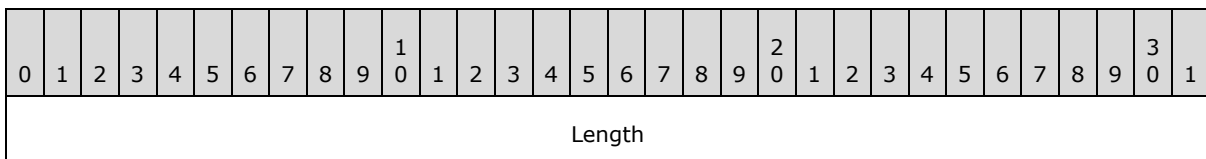
0	1	2	3	4	5	6	7	8	9	1 0	1	2	3	4	5	6	7	8	9	2 0	1	2	3	4	5	6	7	8	9	3 0	1
Length																															

String (variable)																															
...																															

Length (4 bytes): This MUST be set to the number of ANSI characters in the **String** field, including the terminating null character. Length MUST be set to 0x00000000 to indicate an empty string.

String (variable): This MUST be a null-terminated ANSI string.

2.1.5 LengthPrefixedUnicodeString

This structure specifies a **Unicode 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
Length																															
String (variable)																															
...																															

Length (4 bytes): This MUST be set to the number of **Unicode** characters in the **String** field, including the terminating null character. Length MUST be set to 0x00000000 to indicate an empty string.

String (variable): (Optional.) This MUST be a Unicode string.

2.1.6 DEVMODEA

This structure is identical to the **DEVMODE** structure specified in [MS-RPRN] section 2.2.1.1.1, except for the dmDeviceName and dmFormName fields:

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
dmDeviceName																															
dmFormName																															
dmSpecVersion																dmDriverVersion															
dmSize																dmDriverExtra															
dmFields																															
dmOrientation																dmPaperSize															

dmPaperLength	dmPaperWidth
dmScale	dmCopies
dmDefaultSource	dmPrintQuality
dmColor	dmDuplex
dmYResolution	dmTTOption
dmCollate	reserved0
reserved1	
reserved2	
reserved3	
dmNup	
reserved4	
dmICMMethod	
dmICMIntent	
dmMediaType	
dmDitherType	
reserved5	
reserved6	
reserved7	
reserved8	

dmDeviceName (4 bytes): This field is a 32-element array of 8-bit ANSI characters.

dmFormName (4 bytes): This field is a 32-element array of 8-bit ANSI characters.

dmSpecVersion (2 bytes): The version of initialization data specification on which the **DEVMODE** structure is based.

dmDriverVersion (2 bytes): For printers, an optional, implementation-defined version of the **printer driver**.

dmSize (2 bytes): The size, in bytes, of the **DEVMODE** structure. The size MUST NOT include the length of any private, printer driver–specific data that might follow the **DEVMODE** structure's public fields.

dmDriverExtra (2 bytes): The size, in bytes, of the private printer driver data that follows this structure.

dmFields (4 bytes): A bitfield that specifies the fields of the **DEVMODE** structure that have been initialized. If a bit is set, the corresponding field MUST be initialized and MUST be processed on receipt. If a bit is not set, the value of the corresponding field SHOULD be set to zero and MUST be ignored on receipt.

dmOrientation (2 bytes): For printers, the orientation for output. If the DM_ORIENTATION bit is set in **dmFields**, this value MUST be specified.

dmPaperSize (2 bytes): For printers, the size of the output media. If the DM_PAPERSIZE bit is set in **dmFields**, this value MUST be specified. The value of this field SHOULD be one of the following, or it MAY be a device-specific value that is greater than or equal to 0x0100.

dmPaperLength (2 bytes): If the DM_PAPERLENGTH bit is set in the **dmFields** field, the value of this field specifies the length of the paper, in tenths of a millimeter, to use in the printer for which the job is destined.

dmPaperWidth (2 bytes): If the DM_PAPERWIDTH bit is set in the **dmFields** field, the value of this field specifies the width of the paper, in tenths of a millimeter, to use in the printer for which the job is destined.

dmScale (2 bytes): If the DM_SCALE bit is set in the **dmFields** field, the value of this field specifies the percentage factor by which the printed output is to be scaled.

dmCopies (2 bytes): If the DM_COPIES bit is set in the **dmFields** field, the value of this field specifies the number of copies to be printed, if the device supports multiple-page copies.

dmDefaultSource (2 bytes): If the DM_DEFAULTSOURCE bit is set in the **dmFields** field, the value of this field specifies the paper source.

dmPrintQuality (2 bytes): If the DM_PRINTQUALITY bit is set in the **dmFields** field, the value of this field specifies the printer resolution. The value of this field MUST be either a positive value that specifies a device-dependent resolution in dots per inch (DPI) or one of the following four predefined device-independent values that are mapped to a device-specific resolution in an implementation-specific manner.

dmColor (2 bytes): If the DM_COLOR bit is set in the **dmFields** field, the value of this field specifies the color mode to use with color printers.

dmDuplex (2 bytes): If the DM_DUPLEX bit is set in the **dmFields** field, the value of this field specifies duplex or double-sided printing for printers that are capable of duplex printing.

dmYResolution (2 bytes): If the DM_YRESOLUTION bit is set in the **dmFields**, the value of this field specifies the y-resolution, in dots per inch, of the printer.

dmTTOption (2 bytes): If the DM_TTOPTION bit is set in the **dmFields** field, the value of this field specifies how TrueType fonts MUST be printed.

dmCollate (2 bytes): If the DM_COLLATE bit is set in the **dmFields** field, the value of this field specifies whether collation MUST be used when printing multiple copies.

reserved0 (2 bytes): A value that SHOULD be zero and MUST be ignored on receipt.

reserved1 (4 bytes): A value that SHOULD be zero and MUST be ignored on receipt.

reserved2 (4 bytes): A value that SHOULD be zero and MUST be ignored on receipt.

reserved3 (4 bytes): A value that SHOULD be zero and MUST be ignored on receipt.

dmNup (4 bytes): If the DM_NUP bit is set in the **dmFields**, the value of this field specifies the responsibility for performing page layout for **N-Up Printing**.

reserved4 (4 bytes): A value that SHOULD be zero and MUST be ignored on receipt.

dmICMMethod (4 bytes): If the DM_ICMMETHOD bit is set in the **dmFields** field, the value of this field specifies how **Image Color Management (ICM)** is handled. For a non-ICM application, this field determines if ICM is enabled or disabled. For ICM applications, the system examines this field to determine how to handle ICM support. For values see [MS-RPRN]section 2.2.2.1.

dmICMIntent (4 bytes): If the DM_ICMINTENT bit is set in the **dmFields** field, the value of this field specifies which **color matching** method, or intent, MUST be used by default. This field is primarily for non-ICM applications. ICM applications can establish intents by using the ICM functions. For values see [MS-RPRN]section 2.2.2.1.

dmMediaType (4 bytes): If the DM_MEDIATYPE bit is set in the **dmFields** field, the value of this field specifies the type of media to print on. For values see [MS-RPRN]section 2.2.2.1.

dmDitherType (4 bytes): If the DM_DITHERTYPE bit is set in the **dmFields** field, the value of this field specifies how **dithering** is to be done. For values see [MS-RPRN]section 2.2.2.1.

reserved5 (4 bytes): A value that SHOULD be zero and MUST be ignored on receipt.

reserved6 (4 bytes): A value that SHOULD be zero and MUST be ignored on receipt.

reserved7 (4 bytes): A value that SHOULD be zero and MUST be ignored on receipt.

reserved8 (4 bytes): A value that SHOULD be zero and MUST be ignored on receipt.

2.1.7   DVTARGETDEVICE

The DVTARGETDEVICE structure specifies information about a device (such as a display or printer device) that renders the presentation 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
DriverNameOffSet																DeviceNameOffSet															
PortNameOffSet																ExtDevModeOffSet															
DriverName (variable)																															
...																															

DeviceName (variable)
...
PortName (variable)
...
ExtDevMode (variable)
...

DriverNameOffSet (2 bytes): This MUST be set to the offset, in bytes, from the beginning of the structure to the **DriverName** field. If this field is set to 0x0000, the **DriverName** field MUST NOT be present.

DeviceNameOffSet (2 bytes): This MUST be set to the offset, in bytes, from the beginning of the structure to the **DeviceName** field. If this field is set to 0x0000, the **DeviceName** field MUST NOT be present.

PortNameOffSet (2 bytes): This MUST be set to the offset, in bytes, from the beginning of the structure to the **PortName** field. If this field is set to 0x00000, the **PortName** field MUST NOT be present.

ExtDevModeOffSet (2 bytes): This MUST be set to the offset, in bytes, from the beginning of the structure to the **ExtDevMode** field. If this field is set to 0x0000, the **ExtDevMode** field MUST NOT be present. Any gaps between the end of this field and the beginning of the next field MUST be ignored on processing.

DriverName (variable): This MUST be a null-terminated ANSI string that contains a hint on how to display or print presentation data. The creator of this data structure MUST NOT assume that it will be understood during processing. On processing, the hint MAY be ignored. Any gaps between the end of this field and the beginning of the next field MUST be ignored on processing.<3>

DeviceName (variable): This MUST be a null-terminated ANSI string that contains a hint on how to display or print presentation data. The creator of this data structure MUST NOT assume that it will be understood during processing. On processing, the hint MAY be ignored. Any gaps between the end of this field and the beginning of the next field MUST be ignored on processing. This field is optional.<4>

PortName (variable): This MUST be a null-terminated ANSI string that contains any arbitrary value and MUST be ignored on processing. Any gaps between the end of this field and the beginning of the next field MUST be ignored on processing. This field is optional.

ExtDevMode (variable): This MUST contain a DEVMODEA structure (as specified in section 2.1.6). This field is optional.

2.1.8 MetaFilePresentationDataWidth

This MUST be a long value that contains the width of a metafile (as specified in [MS-WMF] section 1.3.1) in logical units. The MM_ANISOTROPIC mapping mode (as specified in [MS-WMF] section 2.1.1.16) MUST be used to convert the logical units to physical units.

2.1.9   MetaFilePresentationDataHeight

This MUST be a long value that contains the height of a metafile (as specified in [MS-WMF] section 1.3.1) in logical units. The **MM_ANISOTROPIC** mapping mode (as specified in [MS-WMF] section 2.1.1.16) MUST be used to convert the logical units to physical units.

2.1.10   EnhancedMetaFilePresentationDataWidth

This MUST be a long value that contains the width of an Enhanced metafile (as specified in [MS-EMF] section 1.3.1) in logical units. The MM_HIMETRIC mapping mode (as specified in [MS-EMF] section 2.1.21) MUST be used to convert the logical units to physical units.

2.1.11   EnhancedMetaFilePresentationDataHeight

This MUST be a long value that contains the height of an enhanced metafile (as specified in [MS-EMF] section 1.3.1) in logical units. The MM_HIMETRIC mapping mode (as specified in [MS-EMF] section 2.1.21) MUST be used to convert the logical units to physical units.

2.1.12   DIBPresentationDataWidth

This MUST be a long value that contains the width of a Device Independent Bitmap object (as specified in [MS-WMF] section 2.2.2.9) in logical units. The MM_HIMETRIC mapping mode (as specified in [MS-WMF] section 2.1.1.16) MUST be used to convert the logical units to physical units.

2.1.13   DIBPresentationDataHeight

This MUST be a **long** value that contains the height of a Device Independent Bitmap object (as specified in [MS-WMF] section 2.2.2.9) in logical units. The MM_HIMETRIC mapping mode (as specified in [MS-WMF] section 2.1.1.16) MUST be used to convert the logical units to physical units.

2.2   OLE1.0 Format Structures

This section specifies the data structures that constitute the OLE1.0 Format.

2.2.1   PresentationObjectHeader

The PresentationObjectHeader structure specifies the header for different types of presentation data structures.

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
OLEVersion																															
FormatID																															
ClassName (variable)																															
...																															

OLEVersion (4 bytes): This can be set to any arbitrary value and MUST be ignored on processing.

FormatID (4 bytes): This MUST be set to 0x00000000 or 0x00000005. If this is set to 0x00000000, the **ClassName** field MUST NOT be present and this structure MUST NOT be contained by another structure. If this is a value other than 0x00000000 or 0x00000005, the PresentationObjectHeader structure is invalid.<5>

Value	Meaning
0x00000000	The **ClassName** field MUST NOT be present.
0x00000005	The **ClassName** field is present.

ClassName (variable): This MUST be a LengthPrefixedAnsiString (section 2.1.4) that identifies the type of the presentation data structure that follows the PresentationObjectHeader.

There are two types of presentation objects. These are specified in sections 2.2.2 and 2.2.3.

2.2.2   StandardPresentationObject

A StandardPresentationObject structure contains a PresentationObjectHeader structure (section 2.2.1). The **ClassName** field of the PresentationObjectHeader MUST be set to the case-sensitive values "METAFILEPICT", "DIB", or "BITMAP".

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
Header (variable)																															
...																															
Width																															
Height																															

Header (variable): This MUST be a PresentationObjectHeader (section 2.2.1). The **FormatID** field of the Header MUST NOT be set to 0x00000000 and the **ClassName** field of the Header MUST be set to the case-sensitive values "METAFILEPICT", "BITMAP", or "DIB". The **ClassName** field identifies the type of the presentation data structure that follows the StandardPresentationObject.

Width (4 bytes): This MUST be set to the width of the presentation object. If the **ClassName** field of the **Header** is set to the case-sensitive value "METAFILEPICT", this MUST be a MetaFilePresentationDataWidth (section 2.1.8). If the **ClassName** field of the **Header** is set to either the case-sensitive value "BITMAP" or the case-sensitive value "DIB", this MUST be a DIBPresentationDataWidth (section 2.1.12).

Height (4 bytes): This MUST be set to the height of the presentation object multiplied by the **LONG** (as specified in [MS-DTYP] section 2.2.27) value -1. If the **ClassName** field of the **Header** is set to the case-sensitive value "METAFILEPICT", this MUST be a MetaFilePresentationDataHeight (section 2.1.9). If the **ClassName** field of the **Header** is set to either the case-sensitive value "BITMAP" or the case-sensitive value "DIB", this MUST be a DIBPresentationDataHeight (section 2.1.13).

There are three types of StandardPresentationObject. These are specified as follows.

2.2.2.1 MetaFilePresentationObject

The MetaFilePresentationObject structure specifies a presentation data type that is used to display linked objects or embedded objects in container applications. The presentation data is in the form of a Windows metafile (as specified in [MS-WMF] section 1.3.1).

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
Header (variable)																															
...																															
PresentationDataSize																															
Reserved1																Reserved2															
Reserved3																Reserved4															
PresentationData (variable)																															
...																															

Header (variable): This MUST be a StandardPresentationObject (section 2.2.2). The **ClassName** field of the contained PresentationObjectHeader (section 2.2.1) MUST be set to the case-sensitive value "METAFILEPICT".

PresentationDataSize (4 bytes): This MUST be an unsigned long integer set to the sum of the size, in bytes, of the **PresentationData** field and the number 8. If this field contains the value 8, the **PresentationData** field MUST NOT be present.

Reserved1 (2 bytes): Reserved. This can be set to any arbitrary value and MUST be ignored on processing.

Reserved2 (2 bytes): Reserved. This can be set to any arbitrary value and MUST be ignored on processing.

Reserved3 (2 bytes): Reserved. This can be set to any arbitrary value and MUST be ignored on processing.

Reserved4 (2 bytes): Reserved. This can be set to any arbitrary value and MUST be ignored on processing.

PresentationData (variable): This MUST be an array of bytes that contain a Windows metafile (as specified in [MS-WMF] section 1.3.1).

2.2.2.2 BitmapPresentationObject

The BitmapPresentationObject structure specifies a presentation data type that is used to display linked objects or embedded objects in container applications. The presentation data is in the form of a Bitmap16 Object structure (as specified in [MS-WMF] section 2.2.2.1).

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
Header (variable)																															
...																															
PresentationDataSize																															
Bitmap (variable)																															
...																															

Header (variable): This MUST be a StandardPresentationObject (section 2.2.2). The **ClassName** field of the contained PresentationObjectHeader (section 2.2.1) MUST be set to the case-sensitive value "BITMAP".

PresentationDataSize (4 bytes): This MUST be an unsigned long integer set to the size, in bytes, of the **Bitmap** field. If this field has the value 0, the **Bitmap** field MUST NOT be present.

Bitmap (variable): This MUST be a Bitmap16 Object structure as specified in [MS-WMF] section 2.2.2.1.

2.2.2.3 DIBPresentationObject

The DIBPresentationObject structure specifies a presentation data type that is used to display linked objects or embedded objects in container applications. The presentation data is in the form of a DeviceIndependentBitmap object structure (as specified in [MS-WMF] section 2.2.2.9).

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
Header (variable)																															
...																															
PresentationDataSize																															
DIB (variable)																															
...																															

Header (variable): This MUST be a StandardPresentationObject (section 2.2.2). The **ClassName** field of the contained PresentationObjectHeader (section 2.2.1) MUST be set to the case-sensitive value "DIB".

PresentationDataSize (4 bytes): This MUST be an unsigned long integer set to the size, in bytes, of the **DIB** field. If this field has the value 0, the **DIB** field MUST NOT be present.

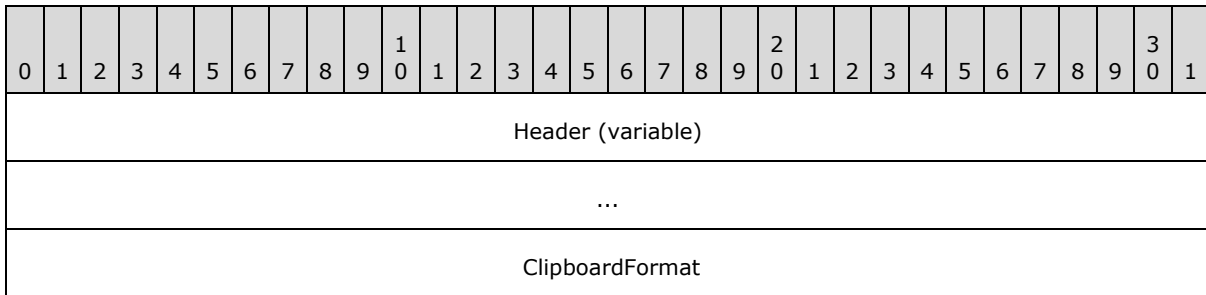**DIB (variable):** This MUST be a DeviceIndependentBitmap Object structure as specified in [MS-WMF] section 2.2.2.9.

2.2.3 GenericPresentationObject

A GenericPresentationObject structure contains a PresentationObjectHeader structure (section 2.2.1). The **ClassName** field of the PresentationObjectHeader MUST NOT be set to "METAFILEPICT", "DIB", or "BITMAP".

There are two types of GenericPresentationObject: the StandardClipboardFormatPresentationObject (section 2.2.3.2) and the RegisteredClipboardFormatPresentationObject (section 2.2.3.3).

2.2.3.1 ClipboardFormatHeader

The ClipboardFormatHeader structure specifies the header for the two types of GenericPresentationObject described 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
Header (variable)																															
...																															
ClipboardFormat																															

Header (variable): This MUST be a PresentationObjectHeader (section 2.2.1). The **FormatID** field of the PresentationObjectHeader MUST NOT be set to 0x00000000 and the **ClassName** field of the **Header** MUST NOT be set to "METAFILEPICT", "DIB", or "BITMAP".

ClipboardFormat (4 bytes): If this is set to 0x0000000, the ClipboardFormatHeader structure MUST be contained by a RegisteredClipboardFormatPresentationObject (see section 2.2.3.3). Otherwise, the ClipboardFormatHeader structure MUST be contained by a StandardClipboardFormatPresentationObject (see section 2.2.3.2). A value other than 0x00000000 MUST identify a standard clipboard format (section 1.3.5.1).

2.2.3.2 StandardClipboardFormatPresentationObject

The StandardClipboardFormatPresentationObject structure specifies a presentation data type that is used to display linked objects or embedded objects in container applications. The presentation data is identified using a standard clipboard format (section 1.3.5.1).
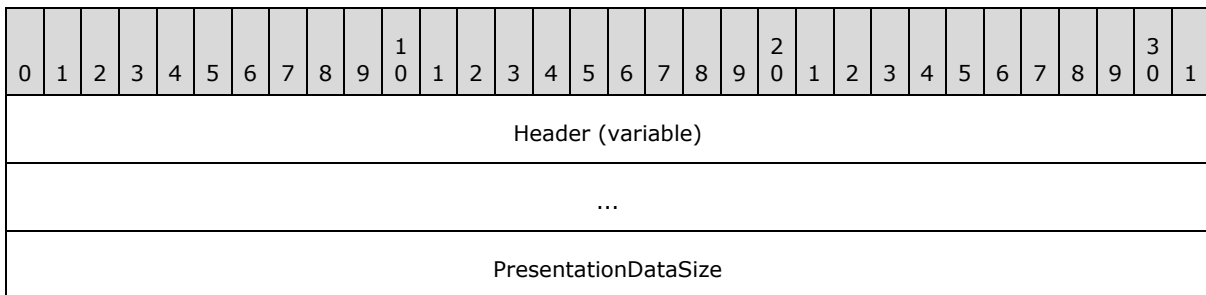
0	1	2	3	4	5	6	7	8	9	1 0	1	2	3	4	5	6	7	8	9	2 0	1	2	3	4	5	6	7	8	9	3 0	1
Header (variable)																															
...																															
PresentationDataSize																															

PresentationData (variable)
...

Header (variable): This MUST be a ClipboardFormatHeader (see section 2.2.3.1). The **ClipboardFormat** field MUST be set to a value other than 0x00000000.

PresentationDataSize (4 bytes): This MUST be an unsigned long integer set to the size, in bytes, of the **PresentationData** field.

PresentationData (variable): This MUST be an array of bytes that contains the presentation data. The format of the data is identified by the Clipboard Format contained in the **Header** field.

If the Clipboard Format contains CF_DIB (section 2.1.1), the **Data** field MUST contain a DeviceIndependentBitmap Object structure (as specified in [MS-WMF] section 2.2.2.9).

If the Clipboard Format contains CF_METAFILEPICT (section 2.1.1), the **Data** field MUST contain a Windows metafile (as specified in [MS-WMF] section 1.3.1). If, after processing the Data field the end of the stream has not been reached, then the **Reserved2** field MUST be present.

If the Clipboard Format contains CF_ENHMETAFILE (section 2.1.1), the **Data** field MUST contain an enhanced metafile (as specified in [MS-EMF] section 1.3.1).

2.2.3.3   RegisteredClipboardFormatPresentationObject

The RegisteredClipboardFormatPresentationObject structure specifies a presentation data type that is used to display linked objects or embedded objects in container applications. The presentation data is identified using a registered Clipboard Format (section 1.7.1).

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
Header (variable)																															
...																															
StringFormatDataSize (optional)																															
StringFormatData (variable)																															
...																															
PresentationDataSize																															
PresentationData (variable)																															
...																															

Header (variable): This MUST be a ClipboardFormatHeader (see section 2.2.3.1). The **ClipboardFormat** field MUST be set to 0x00000000.

StringFormatDataSize (4 bytes): This MUST be set to the size, in bytes, of the **StringFormatData** field.

StringFormatData (variable): This MUST be a LengthPrefixedAnsiString (section 2.1.4) or a LengthPrefixedUnicodeString (section 2.1.5), either of which contain a registered clipboard format name (section 1.7.1).

PresentationDataSize (4 bytes): This MUST be set to the size in, bytes, of the **PresentationData** field.

PresentationData (variable): This MUST be an array of bytes that contains the presentation data that is supplied by an application or a higher-level protocol.

2.2.4 ObjectHeader

The ObjectHeader structure specifies the headers for the LinkedObject (section 2.2.6) and EmbeddedObject (section 2.2.5) structures.

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
OLEVersion																															
FormatID																															
ClassName (variable)																															
...																															
TopicName (variable)																															
...																															
ItemName (variable)																															
...																															

OLEVersion (4 bytes): This can be set to any arbitrary value and MUST be ignored on receipt.

FormatID (4 bytes): This MUST be set to 0x00000001 or 0x00000002. Otherwise, the ObjectHeader structure is invalid.<6>

If this field is set to 0x00000001, the ObjectHeader structure MUST be contained by a LinkedObject structure (see section 2.2.6). If this field is set to 0x00000002, the ObjectHeader structure MUST be contained by an EmbeddedObject structure (see section 2.2.5).

Value	Meaning
0x00000001	The ObjectHeader structure MUST be followed by a LinkedObject structure.
0x00000002	The ObjectHeader structure MUST be followed by an EmbeddedObject structure.

ClassName (variable): This MUST be a LengthPrefixedAnsiString (section 2.1.4) that contains a value identifying the creating application. The value is mapped to the creating application in an implementation-specific manner.<7>

TopicName (variable): This MUST be a LengthPrefixedAnsiString (section 2.1.4).

If the ObjectHeader structure is contained by an EmbeddedObject structure (see section 2.2.5), the TopicName field SHOULD contain an empty string and MUST be ignored on processing.

If the ObjectHeader structure is contained by a LinkedObject structure (see section 2.2.6), the **TopicName** field MUST contain the absolute **path** name of the linked file. The path name either MUST start with a **drive letter** or MUST be in the **Universal Naming Convention (UNC)** format.

ItemName (variable): This MUST be a LengthPrefixedAnsiString (section 2.1.4).

If the ObjectHeader structure is contained by an EmbeddedObject structure (see section 2.2.5), the **ItemName** field SHOULD contain an empty string and MUST be ignored on processing.

If the ObjectHeader structure is contained by a LinkedObject structure (see section 2.2.6),the **ItemName** field MUST contain a string that is used by the application or higher-level protocol to identify the item within the file to which is being linked. The format and meaning of the **ItemName** string is specific to the creating application and MUST be treated by other parties as an opaque string when processing this data structure. An example of such an item is an individual cell within a spreadsheet application.

2.2.5 EmbeddedObject

The EmbeddedObject structure specifies how an embedded object is laid out in a container document.

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
Header (variable)																															
...																															
NativeDataSize																															
NativeData (variable)																															
...																															

Presentation (variable)
...

Header (variable): This MUST be an ObjectHeader (section 2.2.4). The **FormatID** field of the **Header** MUST be set to 0x00000002.

NativeDataSize (4 bytes): This MUST be set to the size of the **NativeData** field, in bytes.

NativeData (variable): This must be an array of bytes that contains the native data.

Presentation (variable): This MUST be a MetaFilePresentationObject (section 2.2.2.1), a BitmapPresentationObject (section 2.2.2.2), a DIBPresentationObject (section 2.2.2.3), a StandardClipboardFormatPresentationObject (section 2.2.3.2), or a RegisteredClipboardFormatPresentationObject (section 2.2.3.3).

2.2.6 LinkedObject

The LinkedObject structure specifies how a linked object is laid out in a container document.

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
Header (variable)																															
...																															
NetworkName (variable)																															
...																															
Reserved																															
LinkUpdateOption																															
Presentation (variable)																															
...																															

Header (variable): This MUST an ObjectHeader structure (section 2.2.4). The **FormatID** field of the **Header** MUST be set to 0x00000001.

NetworkName (variable): This MUST be a LengthPrefixedAnsiString (section 2.1.4).

If the **TopicName** field of the ObjectHeader structure contains a path that starts with a drive letter and if the drive letter is for a remote drive, the **NetworkName** field MUST contain the path name of the linked file in the Universal Naming Convention (UNC) format.

Reserved (4 bytes): This MUST be set to 0x00000000.

LinkUpdateOption (4 bytes): This field contains an implementation-specific hint supplied by the application or higher-level protocol responsible for creating the data structure. The hint MAY be ignored on processing of this data structure.<8>

Presentation (variable): This MUST be a MetaFilePresentationObject (section 2.2.2.1), a BitmapPresentationObject (section 2.2.2.2), a DIBPresentationObject (section 2.2.2.3), a StandardClipboardFormatPresentationObject (section 2.2.3.2), or a RegisteredClipboardFormatPresentationObject (section 2.2.3.3).

2.3 OLE2.0 Format Structures

This section describes the data structures that constitute the OLE2.0 Format. The OLE2.0 Format is specified using Compound File Streams (as specified in [MS-CFB]).

2.3.1 ClipboardFormatOrAnsiString

The ClipboardFormatOrAnsiString structure specifies either a standard clipboard format (section 1.3.5.1) or a registered clipboard format as an ANSI 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
MarkerOrLength																															
FormatOrAnsiString (variable)																															
...																															

MarkerOrLength (4 bytes): If this is set to 0x00000000, the **FormatOrAnsiString** field MUST NOT be present. If this field is set to 0xFFFFFFFF or 0xFFFFFFFE, the **FormatOrAnsiString** field MUST be 4 bytes in size and MUST contain a standard clipboard format identifier (section 2.1.1).

If this set to a value other than 0x00000000, the **FormatOrAnsiString** field MUST be set to a null-terminated ANSI string containing the name of a registered clipboard format (section 2.1.1) and the **MarkerOrLength** field MUST be set to the number of ANSI characters in the **FormatOrAnsiString** field, including the terminating null character.

Value	Meaning
0x00000000	The **FormatOrAnsiString** field MUST NOT be present.
0xfffffffe	The **FormatOrAnsiString** field MUST be 4 bytes in size and MUST contain a standard clipboard format (section 1.3.5.1).
0xffffffff	The **FormatOrAnsiString** field MUST be 4 bytes in size and MUST contain a standard clipboard format (section 1.3.5.1).

FormatOrAnsiString (variable): This MUST be set to a value as specified by the **MarkerOrLength** field.

2.3.2 ClipboardFormatOrUnicodeString

The ClipboardFormatOrUnicodeString structure specifies either a standard clipboard format identifier (section 2.1.1) or a registered clipboard format as a Unicode 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
MarkerOrLength																															
FormatOrUnicodeString (variable)																															
...																															

MarkerOrLength (4 bytes): If this is set to 0x00000000, the **FormatOrUnicodeString** field MUST NOT be present. If this is set to 0xffffffff or 0xfffffffe, the **FormatOrUnicodeString** field MUST be 4 bytes in size and MUST contain a standard clipboard format identifier (section 2.1.1).

Otherwise, the **FormatOrUnicodeString** field MUST be set to a Unicode string containing the name of a registered clipboard format (section 2.1.1) and the **MarkerOrLength** field MUST be set to the number of **Unicode** characters in the **FormatOrUnicodeString** field, including the terminating null character.

Value	Meaning
0xFFFFFFFE	Indicates a standard clipboard format.
0xFFFFFFFF	Indicates a standard clipboard format.
0x00000001 — 0xfffffffd	Indicates a registered clipboard format.

FormatOrUnicodeString (variable): This MUST be set to a value as specified by the **MarkerOrLength** field.

2.3.3 OLEStream

The OLEStream structure is contained inside an **OLE Compound File Stream** object ([MS-CFB] section 1.3). The name of this Compound File Stream object is "\1Ole". The stream object is contained within the **OLE Compound File Storage** object ([MS-CFB] section 1.3) corresponding to the linked object or embedded object (see section 1.3.3). The OLEStream structure specifies whether the storage object is for a linked object or an embedded object. When this structure specifies a storage object for a linked object, it also specifies the reference to the linked 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
Version																															
Flags																															

LinkUpdateOption
Reserved1
ReservedMonikerStreamSize
ReservedMonikerStream (variable)
...
RelativeSourceMonikerStreamSize (optional)
RelativeSourceMonikerStream (variable)
...
AbsoluteSourceMonikerStreamSize (optional)
AbsoluteSourceMonikerStream (variable)
...
ClsidIndicator (optional)
Clsid (optional)
...
...
...
ReservedDisplayName (optional)
Reserved2 (optional)
LocalUpdateTime (optional)
LocalCheckUpdateTime (optional)
RemoteUpdateTime (optional)

Version (4 bytes): This MUST be set to 0x02000001. Otherwise, the OLEStream structure is invalid.<9>

Flags (4 bytes): If this field has a value of 1, the OLEStream structure MUST be for a linked object and the **CLSID** field of the **Compound File Directory Entry** ([MS-CFB] section 2.6.1)

of the **OLE Compound File Storage** object ([MS-CFB] section 1.3) MUST be set to CLSID_StdOleLink ({00000300-0000-0000-C000-000000000046}). If this field has a value of 0, then the OLEStream structure MUST be for an embedded object and the **CLSID** field of the **Compound File Directory Entry** ([MS-CFB] section 2.6.1) of the **OLE Compound File Storage** object ([MS-CFB] section 1.3) MUST be set to the object class GUID of the creating application.

Value	Meaning
0x00000001	The OLEStream structure MUST be for a linked object.
0x00000000	The OLEStream structure MUST be for an embedded object.

LinkUpdateOption (4 bytes): This field contains an implementation-specific hint supplied by the application or by a higher-level protocol that creates the data structure. The hint MAY be ignored on processing of this data structure.<10>

Reserved1 (4 bytes): This MUST be set to 0x00000000. Otherwise, the OLEStream structure is invalid.<11>

ReservedMonikerStreamSize (4 bytes): This MUST be set to the size, in bytes, of the **ReservedMonikerStream** field. If this field has a value 0x00000000, the **ReservedMonikerStream** field MUST NOT be present.

ReservedMonikerStream (variable): This MUST be a MONIKERSTREAM structure (section 2.3.3.1) that can contain any arbitrary value and MUST be ignored on processing.

 Note The fields that follow MUST NOT be present if the OLEStream structure is for an embedded object.

RelativeSourceMonikerStreamSize (4 bytes): This MUST be set to the size, in bytes, of the **RelativeSourceMonikerStream** field. If this field has a value 0x00000000, the **RelativeSourceMonikerStream** field MUST NOT be present.

RelativeSourceMonikerStream (variable): This MUST be a MONIKERSTREAM structure (section 2.3.3.1) that specifies the relative path to the linked object.

AbsoluteSourceMonikerStreamSize (4 bytes): This MUST be set to the size, in bytes, of the **AbsoluteSourceMonikerStream** field. This field MUST NOT contain the value 0x00000000.

AbsoluteSourceMonikerStream (variable): This MUST be a MONIKERSTREAM structure (section 2.3.3.1) that specifies the full path to the linked object.

 If the **RelativeSourceMonikerStream** field is present, it MUST be used by the container application instead of the **AbsoluteSourceMonikerStream**. If the **RelativeSourceMonikerStream** field is not present, the **AbsoluteSourceMonikerStream** MUST be used by the container application.

ClsidIndicator (4 bytes): This MUST be the **LONG** (as specified in section 2.2.27 of [MS-DTYP]) value -1. Otherwise the OLEStream structure is invalid.<12>

Clsid (16 bytes): This MUST be the CLSID (Packet) (section 2.1.2) containing the object class GUID of the creating application.

ReservedDisplayName (4 bytes): This MUST be a LengthPrefixedUnicodeString (section 2.1.5) that can contain any arbitrary value and MUST be ignored on processing.

Reserved2 (4 bytes): This can contain any arbitrary value and MUST be ignored on processing.

LocalUpdateTime (4 bytes): This MUST be a FILETIME (Packet) (section 2.1.3) that contains the time when the container application last updated the **RemoteUpdateTime** field.

LocalCheckUpdateTime (4 bytes): This MUST be a FILETIME (Packet) (section 2.1.3) that contains the time when the container application last checked the update time of the linked object.

RemoteUpdateTime (4 bytes): This MUST be a FILETIME (Packet) (section 2.1.3) that contains the time when the linked object was last updated.

2.3.3.1 MONIKERSTREAM

The MONIKERSTREAM structure specifies the reference to the linked 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
Clsid																															
...																															
...																															
...																															
StreamData (variable)																															
...																															

Clsid (16 bytes): This MUST be the packetized CLSID (section 2.1.2) of an implementation-specific object capable of processing the data contained in the **StreamData** field.

StreamData (variable): This MUST be an array of bytes that specifies the reference to the linked object. The value of this array is interpreted in an implementation-specific manner.<13>

2.3.4 OLEPresentationStream

The OLEPresentationStream structure is contained inside an OLE Compound File Stream object ([MS-CFB] section 1.3) within the OLE Compound File Storage object ([MS-CFB] section 1.3) that corresponds to the linked object or embedded object (see section 1.3.3). There MUST be no more than 999 presentation streams in the storage object. The name of the stream is a concatenation of the prefix "\2OlePres" followed by three numeric characters, each of which is in the range of numbers from '0'-'9'. Some examples of stream names are "\2OlePres000", "\2OlePres123", and "\2OlePres999". The OLEPresentationStream structure specifies the presentation data for linked and embedded objects.

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
AnsiClipboardFormat (variable)																															
...																															
TargetDeviceSize																															
TargetDevice (variable)																															
...																															
Aspect																															
Lindex																															
Advf																															
Reserved1																															
Width																															
Height																															
Size																															
Data (variable)																															
...																															
Reserved2																															
...																															
...																															
...																															
...																	TocSignature															
...																															

AnsiClipboardFormat (variable): This MUST be a ClipboardFormatOrAnsiString structure (section 2.3.1) that contains the Clipboard Format of the presentation data. If the **MarkerOrLength** field of the ClipboardFormatOrAnsiString structure contains 0x0000000, the OLEPresentationStream structure is invalid.<14>

If the **MarkerOrLength** field contains a value other than 0xFFFFFFFF or 0xFFFFFFFE, the value MUST NOT be greater than 0x00000201. Otherwise, the OLEPresentationStream structure is invalid.<15>

If the **FormatOrAnsiString** field of the ClipboardFormatOrAnsiString structure contains the value CF_BITMAP (section 1.3.5.1), the OLEPresentationStream structure is invalid.<16>

TargetDeviceSize (4 bytes): This MUST be set to a value greater than or equal to 0x00000004. If this is set to 0x00000004, the **TargetDevice** field MUST NOT be present. Otherwise, this MUST contain the size of the **TargetDevice** field in bytes.

Value	Meaning
0x00000004	The **TargetDevice** field MUST NOT be present.
0x0000005 — 0xFFFFFFFF	MUST be the size of the **TargetDevice** field in bytes.

TargetDevice (variable): This field MUST contain a DVTARGETDEVICE structure (as specified in section 2.1.7).

Aspect (4 bytes): This field contains an implementation-specific hint on how to render the presentation data on the screen. It MAY be ignored on processing.<17>

Lindex (4 bytes): This field contains an implementation-specific hint on how to render the presentation data on the screen. It MAY be ignored on processing. <18>

Advf (4 bytes): This field contains an implementation-specific hint on how to render the presentation data on the screen. It MAY be ignored on processing.<19>

Reserved1 (4 bytes): This can contain any arbitrary value and MUST be ignored on processing.

Width (4 bytes): This MUST contain the width in pixels of the presentation data.

If the **FormatOrAnsiString** field of **AnsiClipboardFormat** contains CF_DIB (section 2.1.1), this MUST be a DIBPresentationWidth (section 2.1.12).

If the **FormatOrAnsiString** field of **AnsiClipboardFormat** contains CF_METAFILEPICT (section 2.1.1), this MUST be a MetaFilePresentationDataWidth (section 2.1.8).

If the **FormatOrAnsiString** field of **AnsiClipboardFormat** contains CF_ENHMETAFILE (section 2.1.1), this MUST be a EnhancedMetaFilePresentationDataWidth (section 2.1.10).

Height (4 bytes): This MUST contain the height in pixels of the presentation data.

If the **FormatOrAnsiString** field of **AnsiClipboardFormat** contains CF_DIB (section 2.1.1), this MUST be a DIBPresentationDataHeight (section 2.1.13).

If the **FormatOrAnsiString** field of **AnsiClipboardFormat** contains CF_METAFILEPICT (section 2.1.1), this MUST be a MetaFilePresentationDataHeight (section 2.1.9).

If the **FormatOrAnsiString** field of **AnsiClipboardFormat** contains CF_ENHMETAFILE (section 2.1.1), this MUST be a EnhancedMetaFilePresentationDataHeight (section 2.1.11).

Size (4 bytes): This MUST contain the size, in bytes, of the **Data** field.

Data (variable): This MUST contain the presentation data.

If the **FormatOrAnsiString** field of **AnsiClipboardFormat** contains CF_DIB (section 2.1.1), the **Data** field MUST contain a DeviceIndependentBitmap Object structure as specified in [MS-WMF] section 2.2.2.9.

If the **FormatOrAnsiString** field of **AnsiClipboardFormat** contains CF_METAFILEPICT (section 2.1.1), the **Data** field MUST contain a Windows metafile, as specified in [MS-WMF] section 1.3.1. If, after processing the **Data** field the end of the stream has not been reached, then the **Reserved2** field MUST be present.

If the **FormatOrAnsiString** field of **AnsiClipboardFormat** contains CF_ENHMETAFILE (section 2.1.1), the **Data** field MUST contain an Enhanced metafile as specified in [MS-WMF] section 1.3.1.

If the **FormatOrAnsiString** field of **AnsiClipboardFormat** contains the name of a registered clipboard format (section 2.1.1), the format of the **Data** field is specified by the application or the higher level protocol.

Reserved2 (18 bytes): This field MUST be present if the **FormatOrAnsiString** field of **AnsiClipboardFormat** contains CF_METAFILEPICT (section 2.1.1). Otherwise, this field MUST NOT be present. This field can contain any arbitrary value and MUST be ignored on receipt.

TocSignature (4 bytes): If this field does not contain the value 0x494E414E, the **TocEntry** field MUST NOT be present.

TocCount: This MUST contain the number of elements in the **TocEntry** array. If 0, the **TocEntry** structure MUST NOT be present.

TocEntry: This MUST contain an array of TOCENTRY structures (section 2.3.5). The number of structures MUST be specified in the **TocCount** field. The first octet of an array element MUST immediately follow the last octet of the previous element without any alignment gaps.

2.3.5   TOCENTRY

The TOCENTRY structure is used to specify the additional values of the attributes of the OLEPresentationStream structure.

An OLEPresentationStream structure, if present, MUST have one or more values for attributes such as the Clipboard Format and the target device.

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
AnsiClipboardFormat (variable)																															
...																															
TargetDeviceSize																															
Aspect																															
Lindex																															

Tymed
Reserved1
...
...
Advf
Reserved2
TargetDevice (variable)
...

AnsiClipboardFormat (variable): This MUST be a ClipboardFormatOrAnsiString structure (section 2.3.1) containing the Clipboard Format of the presentation data.

TargetDeviceSize (4 bytes): This MUST contain the size, in bytes, of the **TargetDevice** field.

Aspect (4 bytes): This field contains an implementation-specific hint on how to render the presentation data on the screen. It MAY be ignored on processing.<20>

Lindex (4 bytes): This field contains an implementation-specific hint on how to render the presentation data on the screen. It MAY be ignored on processing.<21>

Tymed (4 bytes): This field MUST be ignored on processing.

Reserved1 (12 bytes): This can contain any arbitrary value and MUST be ignored on processing.

Advf (4 bytes): This field contains an implementation-specific hint on how to render the presentation data on the screen. It MAY be ignored on processing.<22>

Reserved2 (4 bytes): This can contain any arbitrary value and MUST be ignored on processing.

TargetDevice (variable): This field MUST contain a DVTARGETDEVICE structure (as specified in section 2.1.7).

2.3.6   OLENativeStream

The OLENativeStream structure is contained inside an **OLE Compound File Stream** object ([MS-CFB] section 1.3). The **OLE Compound File Stream** object is named "\1Ole10Native". The stream object is contained within the OLE Compound File Storage object ([MS-CFB] section 1.3) that corresponds to the linked object or embedded object (see section 1.3.3). This stream is present when native data from a container document in the OLE1.0 format is converted to the OLE2.0 format. The OLENativeStream structure specifies the native data for OLE1.0 embedded objects.

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
NativeDataSize																															
NativeData (variable)																															
...																															

NativeDataSize (4 bytes): This MUST be set to the size, in bytes, of the **NativeData** field.

NativeData (variable): This MUST be set to an array of bytes that contains the native data.

2.3.7 CompObjHeader

The CompObjHeader structure specifies the header of the CompObjStream structure (section 2.3.8).

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
Reserved1																															
Version																															
Reserved2																															
...																															
...																															
...																															
...																															

Reserved1 (4 bytes): This can be set to any arbitrary value and MUST be ignored on processing.

Version (4 bytes): This can be set to any arbitrary value and MUST be ignored on processing.

Reserved2 (20 bytes): This can be set to any arbitrary value and MUST be ignored on processing.

2.3.8 CompObjStream

The CompObjStream structure is contained inside of an **OLE Compound File Stream** (as specified in [MS-CFB]). The **OLE Compound File Stream** has the name "\1CompObj". The CompObjStream structure specifies the Clipboard Format and the display name of the linked object or embedded 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
Header																															
...																															
...																															
...																															
...																															
...																															
...																															
AnsiUserType (variable)																															
...																															
AnsiClipboardFormat (variable)																															
...																															
Reserved1 (variable)																															
...																															
UnicodeMarker (variable)																															
...																															
UnicodeUserType (variable)																															
...																															
UnicodeClipboardFormat (variable)																															
...																															
Reserved2 (variable)																															
...																															

Header (28 bytes): This MUST be a CompObjHeader structure (section 2.3.7).

AnsiUserType (variable): This MUST be a LengthPrefixedAnsiString structure (section 2.1.4) that contains a display name of the linked object or embedded object.

AnsiClipboardFormat (variable): This MUST be a ClipboardFormatOrAnsiString structure (section 2.3.1) that contains the Clipboard Format of the linked object or embedded object. If the **MarkerOrLength** field of the ClipboardFormatOrAnsiString structure contains a value other than 0x00000000, 0xffffffff, or 0xfffffffe, the value MUST NOT be greater than 0x00000190. Otherwise the CompObjStream structure is invalid.<23>

Reserved1 (variable): If present, this MUST be a LengthPrefixedAnsiString structure (section 2.1.4). If the **Length** field of the LengthPrefixedAnsiString contains a value of 0 or a value that is greater than 0x00000028 , the remaining fields of the structure starting with the **String** field of the LengthPrefixedAnsiString MUST be ignored on processing.

If the **String** field of the LengthPrefixedAnsiString is not present, the remaining fields of the structure starting with the **UnicodeMarker** field MUST be ignored on processing.

Otherwise, the **String** field of the LengthPrefixedAnsiString MUST be ignored on processing.

UnicodeMarker (variable): If this field is present and is NOT set to 0x71B239F4, the remaining fields of the structure MUST be ignored on processing.

UnicodeUserType (variable): This MUST be a LengthPrefixedUnicodeString structure (section 2.1.5) that contains a display name of the linked object or embedded object.<24>

UnicodeClipboardFormat (variable): This MUST be a ClipboardFormatOrUnicodeString structure (section 2.3.2) that contains a Clipboard Format of the linked object or embedded object. If the **MarkerOrLength** field of the ClipboardFormatOrUnicodeString structure contains a value other than 0x00000000, 0xffffffff, or 0xfffffffe, the value MUST NOT be more than 0x00000190. Otherwise, the CompObjStream structure is invalid.

Reserved2 (variable): This MUST be a LengthPrefixedUnicodeString (section 2.1.5). The **String** field of the LengthPrefixedUnicodeString can contain any arbitrary value and MUST be ignored on processing.

3   Structure Examples

3.1   OLEStream Structure - Embedded Object

This section describes the binary layout of an OLEStream (section 2.3.3) structure that represents an embedded object.

```
00000000    01 00 00 02 00 00 00 00-00 00 00 00 00 00 00 00   ................
```

Byte offset	Field name	Field value
0x00000000	Version	0x02000001
0x00000004	Flags	0x00000000 (embedded object)
0x00000008	LinkUpdateOption	0x00000000
0x0000000C	Reserved	0x00000000

3.2   OLEStream Structure - Linked Object

This section describes the binary layout of an OLEStream (section 2.3.3) structure that represents a linked object.

```
00000000    01 00 00 02 01 00 00 00-01 00 00 00 00 00 00 00    ................
00000010    00 00 00 00 55 00 00 00-03 03 00 00 00 00 00 00    ....U...........
00000020    c0 00 00 00 00 00 00 46-01 00 09 00 00 00 74 65    .......F......te
00000030    73 74 2E 78 6C 73 00 FF-FF AD DE 00 00 00 00 00    st.xls..........
00000040    00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 16    ................
00000050    00 00 00 10 00 00 00 03-00 74 00 65 00 73 00 74    .........t.e.s.t
00000060    00 2E 00 78 00 6C 00 73-00 65 02 00 00 03 03 00    ...x.l.s.e......
00000070    00 00 00 00 00 C0 00 00-00 00 00 00 46 00 00 18    ............F...
00000080    00 00 00 45 3A 5C 6F 6C-65 64 73 5C 65 78 63 65    ...E:\oleds\exce
00000090    6C 5C 74 65 73 74 2E 78-6C 73 00 FF FF AD DE 00    l\test.xls......
000000A0    00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 02    ................
000000B0    00 00 00 17 02 00 00 2E-00 00 00 03 00 45 00 3A    .............E.:
000000C0    00 5C 00 6F 00 6C 00 65-00 64 00 73 00 5C 00 65    .\.o.l.e.d.s.\.e
000000D0    00 78 00 63 00 65 00 6C-00 5C 00 74 00 65 00 73    .x.c.e.l.\.t.e.s
000000E0    00 74 00 2E 00 78 00 6C-00 73 00 DD 01 00 00 05    .t...x.l.s......
000000F0    00 4C 00 00 00 01 14 02-00 00 00 00 00 c0 00 00    .L..............
00000100    00 00 00 00 46 83 00 00-00 20 08 00 00 7d 69 d0    ....F.... ...}i.
00000110    31 C4 AF C8 01 D6 CB D2-31 C4 AF C8 01 E0 5B 7E    1.......1.....[~
00000120    A2 C4 AF C8 01 00 58 00-00 00 00 00 00 01 00 00    ......X.........
00000130    00 00 00 00 00 00 00 00-00 00 00 00 00 db 00 14    ................
00000140    00 1F 50 E0 4F D0 20 EA-3A 69 10 A2 D8 08 00 2B    ..P.O. .:i.....+
00000150    30 30 9D 19 00 2F 45 3A-5C 00 00 00 00 00 00 00    00.../E:\.......
00000160    00 00 00 00 00 00 00 00-00 00 00 00 36 00 31 00    ............6.1.
00000170    00 00 00 00 A6 38 ED 9E-10 08 6F 6C 65 64 73 00    .....8....oleds.
00000180    22 00 03 00 04 00 EF BE-8C 38 78 0E A6 38 ED 9E    "........8x..8..
00000190    14 00 00 00 6F 00 6C 00-65 00 64 00 73 00 00 00    ....o.l.e.d.s...
000001A0    14 00 36 00 31 00 00 00-00 a6 38 08 B3 10 08    ..6.1......8....
000001B0    65 78 63 65 6C 00 22 00-03 00 04 00 EF BE 99 38    excel."........8
000001C0    B2 0A A6 38 08 B3 14 00-00 65 00 78 00 63 00    ...8.....e.x.c.
000001D0    65 00 6C 00 00 00 14 00-40 00 32 00 00 58 00    e.l.....@.2..X..
000001E0    A6 38 37 B3 20 08 74 65-73 74 2E 78 6C 73 00 00    .87. .test.xls..
000001F0    28 00 03 00 04 00 EF BE-A6 38 33 AF A6 38 33 AF    (.......83..83.
00000200    14 00 00 00 74 00 65 00-73 00 74 00 2E 00 78 00    ....t.e.s.t...x.
00000210    6C 00 73 00 00 00 18 00-00 00 50 00 00 00 1C 00    l.s.......P.....
00000220    00 00 01 00 00 00 1C 00-00 00 37 00 00 00 00 00    .........7.....
00000230    00 00 4F 00 00 00 1B 00-00 00 03 00 00 00 E1 99    ..O............
00000240    1E 18 10 00 00 00 4E 65-77 20 56 6F 6C 75 6D 65    ......New Volume
00000250    00 45 3A 5C 6F 6C 65 64-73 5C 65 78 63 65 6C 5C    .E:\oleds\excel\
00000260    74 65 73 74 2E 78 6C 73-00 00 60 00 00 00 03 00    test.xls..`.....
00000270    00 A0 58 00 00 00 00 00-00 00 78 78 78 78 78 78    ..X.......xxxxxx
00000280    78 78 78 00 00 00 00 00-00 00 B6 F5 56 97 AB 4A    xxx.........V..J
00000290    6A 40 B6 b1 84 E1 8D D1-76 EE AC 14 D9 7D 90 18    j@......v....}..
000002A0    DD 11 BA D5 00 0B DB CA-27 8E B6 F5 56 97 Ab 4A    ........'..V..J
000002B0    6A 40 B6 B1 84 E1 8D D1-76 EE AC 14 D9 7D 90 18    j@......v....}..
000002C0    DD 11 BA D5 00 0B DB CA-27 8E 00 00 00 00 FF FF    ........'.......
000002D0    FF FF 20 08 02 00 00 00-00 00 c0 00 00 00 00 00    .. ............
000002E0    00 46 00 00 00 00 FF FF-FF FF F0 4E 44 26 C8 AF    .F.........ND&..
000002F0    C8 01 60 B9 45 27 C8 AF-C8 01 F0 4e 44 26 C8 AF    ..`.E'.....ND&..
00000300    C8 01                                              ..
```

Byte offset	Field name	Field value
0x00000000	Version	0x02000001

Byte offset	Field name	Field value
0x00000004	Flags	0x00000001 (linked object)
0x00000008	LinkUpdateOption	0x00000001
0x0000000C	Reserved	0x00000000
0x00000010	ReservedMonikerStreamSize	0x00000000
0x00000014	RelativeSourceMonikerStreamSize	0x00000055
0x00000018	RelativeSourceMonikerStream.Clsid	{00000303-0000-0000-C000-000000000046}
0x00000028	RelativeSourceMonikerStream.StreamData	
0x00000069	AbsoluteSourceMonikerStreamSize	0x00000265
0x0000006D	AbsoluteSourceMonikerStream.Clsid	{00000303-0000-0000-C000-000000000046}
0x0000007D	AbsoluteSourceMonikerStream. StreamData	
0x000002CE	ClsidIndicator	0xFFFFFFFF
0x000002D2	Clsid	{00020820-0000-0000-c000-000000000046}
0x000002E2	ReservedDisplayName.Length	0x00000000
0x000002E6	Reserved2	0xFFFFFFFF
0x000002EA	LocalUpdateTime.dwLowDateTime	0x26444EF0
0x000002EE	LocalUpdateTime.dwHighDateTime	0x01C8AFC8 (05/06/2008 15:25:55:039)
0x000002F2	LocalCheckUpdateTime.dwLowDateTime	0x2745B960
0x000002F6	LocalCheckUpdateTime.dwHighDateTime	0x01C8AFC8 (05/06/2008 15:25:56:726)
0x000002FA	RemoteUpdateTime.dwLowDateTime	0x26444EF0
0x000002FE	RemoteUpdateTime.dwHighDateTime	0x01C8AFC8 (05/06/2008 15:25:55:039)

3.3   OLEPresentationStream Structure

This section describes the binary layout of the OLEPresentationStream (section 2.3.4) structure.

```
00000000      FF FF FF FF 08 00 00 00-04 00 00 00 01 00 00 00    ................
00000010      FF FF FF FF 02 00 00 00-00 00 00 00 91 74 00 00    .............t..
00000020      A7 42 00 00 40 4E 21 00-28 00 00 00 68 04 00 00    .B..@N!.(...h...
00000030      85 02 00 00 01 00 18 00-00 00 00 00 18 4E 21 00    .............N!.
00000040      C4 0E 00 00 C4 0E 00 00-00 00 00 00 00 00 00 00    ................
00000050      FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF    ................
00000060      FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF    ................
00000070      FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF    ................
...           ...
00214E58      FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF    ................
00214E68      4E 41 4E 49 01 00 00 00-FF FF FF FF 03 00 00 00    NANI............
00214E78      00 00 00 00 01 00 00 00-FF FF FF FF 20 00 00 00    ............ ...
00214E98      95 74 00 00 AA 42 00 00-16 00 00 00 02 00 00 00    .t...B..........
00214EA8      18 00 00 00                                        ....
```

Byte offset	Field name	Field value
0x00000000	AnsiClipboardFormat.MarkerOrLength	0xFFFFFFFF
0x00000004	AnsiClipboardFormat.FormatOrAnsiString	0x00000008 (CF_DIB)
0x00000008	TargetDeviceSize	0x00000004
0x0000000C	Aspect	0x00000001
0x00000010	Lindex	0xFFFFFFFF
0x00000014	Advf	0x00000002
0x00000018	Reserved1	0x00000000
0x0000001C	Width	0x00007491
0x00000020	Height	0x000042A7
0x00000024	Size	0x00214E40
0x00000028	Data	
...	Data	
0x00214E68	TocSignature	0x494E414E
0x00214E6C	TocCount	0x00000001

3.4 TOCENTRY Structure

This section describes the binary layout of the TOCENTRY (section 2.3.5) structure.

```
00000000      FF FF FF FF 03 00 00 00-00 00 00 00 01 00 00 00    ................
00000010      FF FF FF FF 20 00 00 00-95 74 00 00 AA 42 00 00    .... ....t...B..
00000020      16 00 00 00 02 00 00 00-18 00 00 00                ............
```

Byte offset	Field name	Field value
0x00000000	AnsiClipboardFormat.MarkerOrLength	0xFFFFFFFF
0x00000004	AnsiClipboardFormat.FormatOrAnsiString	0x00000003 (CF_METAFILEPICT)
0x00000008	TargetDeviceSize	0x00000000
0x0000000C	Aspect	0x00000001
0x00000010	Lindex	0xFFFFFFFF
0x00000014	Tymed	0x00000020
0x00000018	Reserved1[0]	0x00007495
0x0000001C	Reserved1[1]	0x000042AA
0x00000020	Reserved1[2]	0x00000016
0x00000024	Advf	0x00000002
0x00000028	Reserved2	0x00000018

3.5 OLENativeStream Structure

This section describes the binary layout of the OLENativeStream (section 2.3.6) structure.

```
00000000      80 F3 01 00 42 4D 6E F3-01 00 00 00 00 00 36 00    ....BMn.......6.
00000010      00 00 28 00 00 00 C8 00-00 00 D5 00 00 00 01 00    ..(.............
00000020      18 00 00 00 00 00 38 F3-01 00 00 00 00 00 00 00    ......8.........
00000030      00 00 00 00 00 00 00 00-00 00 FF FF FF FF FF FF    ................
00000040      FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF    ................
...           ...
```

Byte offset	Field name	Field value
0x00000000	NativeDataSize	0x0001F380
0x00000004	NativeData	

4 Security Considerations

None.

Release: Friday, February 4, 2011

5   Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

- Microsoft Windows NT® 4.0 operating system

- Microsoft 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

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product 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 1.7: Windows only uses the values in [MS-ERREF].

<2> Section 2: The Windows implementation of this structure uses the default system code page as the ANSI code page for this structure. For more information, see [MSDN-CP].

<3> Section 2.1.7: In Windows, this field contains the name of the display or the printer driver as described in [MSDN-CREATEDC].

<4> Section 2.1.7: In Windows, this field contains the name of the display or the printer driver as described in [MSDN-CREATEDC].

<5> Section 2.2.1: Windows will fail processing a PresentationObjectHeader structure with error CONVERT10_E_OLESTREAM_FMT (as specified in [MS-ERREF]) if the **FormatID** field is a value other than 0x00000000 or 0x00000005.

<6> Section 2.2.4: Windows will fail to process an ObjectHeader structure with error CONVERT10_E_OLESTREAM_FMT (as specified in [MS-ERREF]) if the **FormatId** field is a value other than 0x00000001 or 0x00000002.

<7> Section 2.2.4: In Windows the **ClassName** field contains the ProgID (see [MSDN-ProgID]) of a Component Object Model (COM) component (see [MSDN-COM]) belonging to the creating application.

<8> Section 2.2.6: In Windows, this field contains values from the **OLEUPDATE** enumeration as described in [MSDN-OLEUPDATE].

<9> Section 2.3.3: Windows will fail processing an OLEStream structure with error DV_E_CLIPFORMAT (as specified in [MS-ERREF]) if the value of the **Version** field is not 0x02000001.

<10> Section 2.3.3: In Windows, this field contains values from the OLEUPDATE enumeration as described in [MSDN-OLEUPDATE].

<11> Section 2.3.3: Windows will fail to process such a structure with error DV_E_CLIPFORMAT (as specified in [MS-ERREF]) if the value of this field is not 0x02000001.

<12> Section 2.3.3: Windows will fail processing such a structure with the error E_FAIL (as specified in [MS-ERREF]) if this field contains a value other than the **LONG** (as specified in section 2.2.27 of [MS-DTYP]) value -1.

<13> Section 2.3.3.1: In Windows, the CLSID present in the **StreamData** field is that of a **COM** (see [MSDN-COM]) object that implements the **IPersistStream** interface (see [MSDN-IPersistStream]) On receipt, Windows activates the COM object, obtains the **IPersistStream** interface and calls the **Load** method of the interface passing an **IStream** object reference (see [MSDN-IStream]) that encapsulates the **StreamData** field.

<14> Section 2.3.4: Windows will fail to process the structure with the error DV_E_CLIPFORMAT (as specified in [MS-ERREF]) if the **MarkerOrLength** field contains 0x0000000.

<15> Section 2.3.4: Windows will fail to process the structure with the error DV_E_CLIPFORMAT (as specified in [MS-ERREF]) if the **MarkerOrLength** field contains a value other than 0xffffffff or 0xfffffffe and the value is greater than 0x00000201.

<16> Section 2.3.4: Windows will fail to process the structure with the error DV_E_CLIPFORMAT (as specified in [MS-ERREF]) if the **FormatOrAnsiString** field contains the value CF_BITMAP.

<17> Section 2.3.4: In Windows, this field contains values from the **DVASPECT** enumeration as described in [MSDN-DVAPSECT].

<18> Section 2.3.4: In Windows, this field contains the lindex value as described in [MSDN-FRMATCLNDX].

<19> Section 2.3.4: In Windows, this field contains values from the **ADVF** enumeration as described in [MSDN-ADVF].

<20> Section 2.3.5: In Windows, this field contains values from the **DVASPECT** enumeration as described in [MSDN-DVAPSECT]

<21> Section 2.3.5: In Windows, this field contains the FORMATETCLINDEX value as described in [MSDN-FRMATCLNDX].

<22> Section 2.3.5: In Windows, this field contains values from the **ADVF** enumeration as described in [MSDN-ADVF].

<23> Section 2.3.8: Windows will fail to process the structure with the error DV_E_CLIPFORMAT (as specified in [MS-ERREF]) if the **MarkerOrLength** field contains a value other than 0x00000000, 0xffffffff, or 0xfffffffe and the value is greater than 0x00000190.

<24> Section 2.3.8: Windows will fail to process the structure with the error **DV_E_CLIPFORMAT** (as specified in [MS-ERREF]) if the **MarkerOrLength** field of the LengthPrefixedUnicodeString structure contains a value other than 0x00000000, 0xffffffff, or 0xfffffffe and the value is greater than 0x00000190.

6 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

7 Index