Inspection and Sanitization Guidance for Microsoft Office 2003

Version 1.0.2

17 August 2010

National Security Agency
9800 Savage Rd, Suite 6721
Ft. George G. Meade. MD 20755

Authored/Released by:
Unified Cross Domain Capabilities Office
cds_tech@nsa.gov

DOCUMENT CHANGE HISTORY

Date	Version	Description
08/17/2010	1.0.2	Initial Release
12/13/2017	1.0.2	Updated Contact information, IAC Logo, Cited Trademarks and Copyrights, Expanded Acronyms, and added Legal Disclaimer

DISCLAIMER OF WARRANTIES AND ENDORSEMENT

The information and opinions contained in this document are provided "as is" and without any warranties or guarantees. References herein to any specific commercial product, process, or service by trade name, trademark, manufacturer or otherwise, does not constitute or imply its endorsement, recommendation or favoring by the United States Government and this guidance shall not be used for advertising or product endorsement purposes.

EXECUTIVE SUMMARY

This *Inspection and Sanitization Guidance for Microsoft Office 2003* document provides guidance and specifications for developing file inspection and sanitization software for Microsoft® Office 2003 files (i.e., word processing, presentation, and spreadsheet documents).

Client programs, such as Microsoft® Word 2003, can store arbitrary data in an Office 2003 document, including video, sounds, and hidden text. This presents challenges for automated file processing software. This document addresses this complexity by delineating the various constructs with the Office 2003 file formats, and it provides specific guidance for designing, building, and testing Office 2003 file inspection and sanitization applications.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. SCOPE

1.1 Purpose of this Document

This *Inspection and Sanitization Guidance (ISG) for Microsoft Office 2003* document outlines the findings of the Cross Domain Solutions (CDS) team's research into potential areas of concern regarding the binary data format that Microsoft® (MS) Corporation uses in its Office 2003 product suite (i.e., Excel, PowerPoint and Word) [1].

The intended audience for this document includes system engineers, designers, software developers and testers who work on file inspection and sanitization applications that involve processing Office 2003 files.

1.2 Introduction

The frequent exchange of MS Office documents between federal government entities and business partners creates a significant potential risk for releasing sensitive information. In this context, sensitive information falls into two general categories: unintentional release of data and intentional obfuscation of data to hide information in a file. This document focuses on the accidental release of data and sanitization of hidden data in MS Office 1997-2003 documents and provides guidance for accomplishing these tasks.

The MS Office 1997-2003 file format is a layered format that contains several abstractions to facilitate storage of arbitrary data. At the lowest level, the file format functions as a file system that provides storage entities (sectors) to hold data for higher levels. The ability to store arbitrary data in the format gives users great flexibility and functionality; however, it also increases the risk for accidental data leakage and potential for hidden data, including execution code.

1.3 Background

This research effort grew out of a CDS filter modularization prototype effort for MS Office 2007 files. During rapid development of an Office 2007 filter, the team quickly realized that they, system designers, testers, and users of file inspection and sanitization applications required a firm understanding of file format constructs (i.e., internal file format data structures). This realization led to the development of guidance for Office

[1] Microsoft, Excel, PowerPoint, Word and MS Office are registered trademarks of Microsoft Corporation

2003 and Office 2007 file formats, and an expectation of future guidance for other file formats.

The goal is for this guidance to serve as a checklist and reference guide for file format constructs, especially constructs that are considered areas of concern for file inspection and sanitization programs. This guidance does not substitute for the official reference documentation for the file formats, which are considered authoritative.

The team also produced a companion document to this guidance—a spreadsheet called the *Data Transfer Guidance (DTG)*. The DTG template outlines each construct in this ISG document along with recommended actions for handling each construct. Administrators can use the template to specify actions that a file inspection or sanitization program should implement.

1.4 Document Organization

This document describes many, but not all, data structures that MS Office 2003 files use. It divides data structures into two major sections: a section for the underlying file structure and client specific data structures and a section that describes metadata and other high level data elements. The document formally defines each data structure. Each definition contains recommendations that specify how inspection and sanitization programs can handle the data structure.

Table 1-1 summarizes the organization of this document.

Table 1-1. Document Organization

Section	Description
Section 1: Scope	This section describes the purpose of this document, project background, document organization, recommendation actions, and scope limitations of this document.
Section 2: Constructs and Taxonomy	This section provides an overview of the MS Office 2003 constructs and taxonomy used in this document.
Section 3: Office 2003 File Structure Overview	This section provides an overview of the file structures, constructs, and metadata that Office 2003 documents use.
Section 4.: Office 2003 Constructs and Metadata	This section details the MS Office 2003 constructs, and associated concerns, and recommendations.
Section 5.: Acronyms	This section lists the acronyms that appear in

Section	Description
	this document.
Section 6: Referenced Documents	This section lists the publications and Web sites that document references.
Section 7: Validating CFB Files	This section provides an overview for validating CFB files.
Section 8: List of Field Codes	This section lists the field codes that appear in this document.
Appendix A: BNF Constructs	This appendix contains BNF grammars for applicable constructs in this document.

1.5 Recommendations

The following subsections summarize the recommendation action categories that appear in this document, associated action options, and naming conventions that denote to which MS Office products recommendations apply.

1.5.1 Actions

Each construct description lists recommended actions for handling the construct when processing a document. Generally, inspection and sanitization programs will perform one or more actions on a construct: *Validate*, *Remove*, *Replace*, *External Filtering Required*, or *Review*.

The Recommendation section in each construct lists each of these actions and applicable explanations. It notes if a particular action does not apply and indicates actions that are not part of the standard set of actions (listed above). For example, a program may choose to reject an encrypted file. Additionally, for some constructs, an action may break down further to specific construct elements to give administrators the flexibility to handle specific elements differently (e.g., for hidden data in Excel, the recommendations for *Remove* include an action for removing hidden sheets and another for removing hidden cells). For many constructs, the recommended actions apply across the Office product line. For other constructs, the recommendations are broken out for each product.

NOTE

The recommendations in this document are brief explanations rather than a How-To Guide. Readers should refer to the construct description or MS Corporation's official documentation for additional details.

Table 1-2 summarizes the recommendation actions.

Table 1-2. Recommendation Actions

Recommendation Action	Comments
Validate	Verify the data structure's integrity, which may include integrity checks on other components in the file. (This should almost always be a recommended action.)
Replace	Replace the data structure or one or more of its elements with values that alleviate the risk (e.g., replacing a user name with a non-identifying, harmless value or just substituting a common name for all authors).
Remove	Remove the data structure or one or more of its elements and any other affected area.
External Filtering Required	Note the data type and pass data to an external action for handling that data type (e.g., extract text and pass it to a dirty word search).
Review	Present the data structure or its constructs for a human to review.

NOTE

No recommendations for logging all actions and found data are included here because all activity logging in a file inspection application should occur "at an appropriate level" and presented in a form that a human can analyze further (e.g., the audit information may be stored in any format but must be parsable and provide enough information to address the issue when presented to a human.)

1.5.2 Action Options

The *DTG* companion to this ISG document specifies four options for each recommended action: *Mandatory*, *Required*, *Optional*, or *Ignore*. Depending on the circumstances (e.g., a low to high data transfer versus a classified to transfer), programs can be configured to handle constructs differently.

Table 1-3 summarizes the recommendation action options

Table 1-3. Recommendation Action Options

Action Options	Comments
Mandatory	For the given direction (e.g., secure private network to unsecure Internet), the file inspection and sanitization program must perform this recommended action.
Recommended	Programs should implement this action if technically feasible.
Optional	Programs may choose to perform or ignore this recommended action.
Ignore	Programs can ignore this data structure entirely

1.5.3 Naming Convention for Recommendations

Recommendations in this document are numbered sequentially, where applicable, and adhere to the following naming conventions:

- ER = MS Excel Recommendation
- PR = MS PowerPoint Recommendation
- WR = MS Word Recommendation
- AR = Recommendations that apply to Excel, PowerPoint, and Word (all).

1.6 DTG Document

Each format documented for inspection and sanitization programs has a companion document (i.e., the aforementioned *DTG* document). The *DTG* serves as a checklist for administrators and others to describe expected behaviors for inspection and sanitization programs. For instance, an administrator may decide to remove all hidden sheets in an Excel spreadsheet but leave hidden cells intact. Or the administrator may decide to remove all hidden data if the document is being transferred to a lower security domain.

The *DTG* gives the administrator the flexibility to specify behaviors for inspection and sanitization programs. The workbook contains a worksheet for each security domain (i.e., the originating domain). Each worksheet lists the numbered constructs from this document and enumerated recommendations in a row. After the recommendations, the worksheet displays a cell for each possible destination domain. This enables an administrator to select the action option for data transfer from the originating domain to the particular destination domain. Each construct row also contains two comment cells: one for low to high transfers and another for high to low transfers.

The recommended actions address two broad risk types: data hiding and data execution. Most data structures are vulnerable to one type of risk, while others are susceptible to both risk types. Each construct row in the *DTG* worksheet contains a cell for designating the risk type (i.e., data execution, data hiding, or both) and another cell for assessing the risk level for that construct (i.e., high, medium and low). This enables administrators to assign the risk type and level to each specific construct.

1.7 Document Limitations

The following subsections describe the scope limitations for this document pertaining to constructs, covert channel analysis, and character encoding.

1.7.1 Constructs

Although this document delves into many low level constructs within Office 2003 documents, it does not serve as a complete reference to all of the different constructs in the Office 2003 formats. This document covers the overall file format, product- specific constructs that govern the specific file formats (e.g., .doc, .xls, and .ppt) and metadata elements. The team identified these as particular concern areas for file inspection and sanitization program developers.

Microsoft recently published its file format specifications. The team *highly* recommends that file inspection and sanitization software developers use this data vigorously. Further, if conflicts exist between this document and Microsoft's specifications, consider the latter authoritative. The documents are available at the following Web site:

https://msdn.microsoft.com/en-us/library/dd208104(PROT.10).aspx

In addition to these extensive documents, much of the research for the guidance in this ISG relied on investigations into the file format that others performed and have openly published. In some cases, they uncovered discrepancies in the official documentation. In these cases, the guidance in this ISG uses the observed findings.

The Open Office project (http://www.openoffice.org/), an open source Office application suite, has investigated the Office 92-2003 file format extensively to ensure compatibility between its products and the MS Office suite. The project published a paper entitled *OpenOffice.org's Documentation of the Microsoft Compound Document File Format*, which details many parts of the Office 92-2003 file format. However, differences exist between this document and the official specifications. Therefore, readers of this ISG should use all available resources and rely on their best judgments and research when decomposing the Office 92-2003 file formats.

Recommendations that appear within each metadata construct in this document are "alternatives"—i.e., available choices that an administrator can make when determining how to handle a specific data structure in a specific application. So, some recommendations are better suited than others in different situations and some recommendations may be more difficult to implement. Also, some complementary recommendations can be grouped together (e.g., "Remove headers" and "Remove footers"). Still, other recommendations may seem contradictory (e.g., "Remove Comments" and "Replace Comments"). See Section 4 for details on the recommendations, actions, and options.

Often, the content of a metadata construct will detail some of the underlying data structures of interest. However, this document is not a "how-to guide". The locations, recommendations, and examples do not imply that the team covered every avenue for identifying metadata. Office 1997-2003 file structures are very complex. Therefore, records contained within them could be interdependent in ways that are not immediately apparent; and removal or alteration of those data structures could have a "cascading effect" on other data structures within the file.

1.7.2 Covert Channel Analysis

It is impossible to identify all available covert channels in the aforementioned file formats. Because Office documents contain free-form text, searching for hidden data becomes increasingly more difficult. No tool can possibly analyze every channel, so this document highlights the highest risk areas to reduce or eliminate data spills and malicious content. Additionally, tool develops should closely follow the integrity rules outlined in Section 7, which can eliminate native covert channels that exploit free space in unused components of the file format.

This document also does not discuss steganography within block text or media files, such as a hidden message that is embedded within an innocuous image or paragraph. Separate file format filters that specialize in steganography should handle embedded content, such as text, images, videos and audio.

1.7.3 Character Encoding Use and Concern

Characters used in words and sentences are grouped into a character set. A numeric representation, or code point of one or more bytes, identifies each character in the character set. Character encoding refers to the mapping used between the code points and the actual characters.

MS Office is based on the Unicode text encoding standard. The Unicode text standard, implemented in Office, supports 16-bit extensible international character coding that covers the world's major languages. The Unicode standard also supports bi-directional text, also known as right-to-left (RTL) and left-to-right (LTR) text. Common scripts that use RTL text include Hebrew, Persian and Arabic.

When modifying an Office 2003 document, use care when addressing different languages and text orientations. Modifications made to embedded content should remain in the original character encoding.

In addition to the issues surrounding character encoding and text direction, consider the use of fonts, such as Wingdings, Webdings, and glyphs mapped to keys, when filtering a document. Although glyphs lose their original key mappings after document saves, they still pose a threat by concealing obfuscated text.

2. CONSTRUCTS AND TAXONOMY

2.1 Constructs

This document attempts to formalize data structure definitions and the potential risks of these data structures by describing each element (hereafter referred to as a "construct") in a precise and predictable format. Each construct definition contains the following information:

- **Description**: a high level explanation of the data structure or element
- **Concern**: an explanation of potential problems this element poses. For example, some metadata elements can cause inadvertent data leakage and others may be used for data exfiltration
- **Location**: provides a textual description of where to find the element in the document (file). This can vary by client (or product) or it may be the same across the entire product line
- **Examples**: if applicable, the definition will contain a construct example
- **Recommendations**: described in Section 4.

The construct is formally defined in a "grammar", as shown in Appendix A. formalization attempts to bring a precision level to the construct's definition and can serve as guidance for developers as they write code to parse and handle the data structure. However, the team strongly encourages developers to not apply the grammar blindly and use to standard software engineering practices and vigorous testing to ensure program correctness.

The start of each construct in this document is denoted via a numbering schema that adheres to the syntax: **OFFICE 2003.x.x: Title,** where "**x**" represents a sequential index and "**Title**" denotes the title of the construct. The end of each construct is denoted via the syntax: **OFFICE 2003.x.x: END**.

2.2 Taxonomy

The following table describes the terms that appear in this document.

Table 2-1 Taxonomy Used in this Document

Term	Definition
Consistency	A state where the construct's internal attributes are set to correct values (as determined by the grammar and/or MS specifications) and all data types within the construct are correctly identified
Construct	A data structure, a metadata object, or other element that holds and organizes data in the MS Office 2003 format
DTG	A spreadsheet that lists all constructs in an ISG and associated recommendations. A DTG is used to define policies for handling every ISG construct when performing inspection and sanitization
Grammar	A precise, formal, and rigorous syntax for defining constructs
Inspection and Sanitization	Activities for processing files to prevent inadvertent data leakage, data exfiltration, and transmission of malicious data or code.
ISG	A document (such as this one) that details a file format or protocol, and inspection and sanitization activities for constructs within that file format or protocol
Recommendations	A series of actions for handling a construct when performing inspection and sanitization activities
Referential Integrity	A construct state where all associated objects are referenced properly in the construct and entries in the construct reference existing objects

3. OFFICE 2003 FILE STRUCTURE OVERVIEW

Office 2003 documents use an object linking and embedding (OLE) common binary format to store files, called the Compound File Binary (CFB) format. A CFB file resembles a traditional file system. Its data objects are stored as files within the file and directory tables that provide reference information to the objects. The CFB gives clients the ability to store arbitrary data easily, such as text, images, video, other files, and executables. Flexibility in the CFB also leads to potential issues, such as data hiding and exploitation attack vectors.

In OLE CFB parlance, directories are referred to as "storages" and file objects are called "streams".

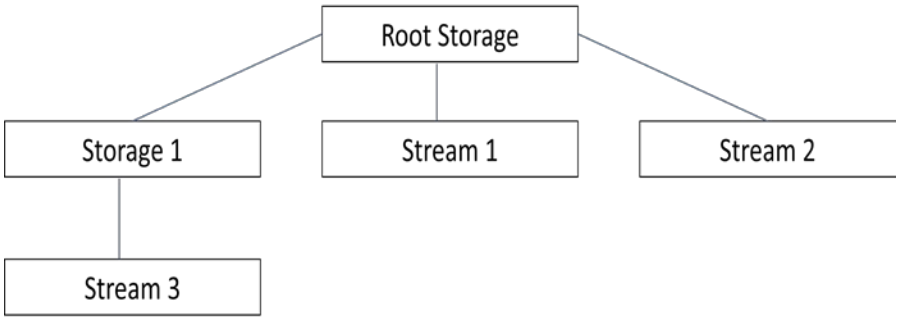

Figure 3-1. Simple CFB File Layout

Each Office application then layers its data on top of the CFB. For example, MS Word layers internal data structures in CFB sectors.

Section 3.1 details the CFB. Section 3.2 discusses OLE. Subsequent sections delineate the storage strategy that client applications use.

3.1 CFB

The CFB contains structures to organize data objects that are stored within the file. Like the MS-DOS® file system, the primary CFB structure is the File Allocation Table (FAT)® [2], which essentially is an index to each individual file system sector, or in the case of this discussion, the file.

The file is divided into sectors of either 512 or 4096 bytes in length. The first sector *always* contains the CFB header, which provides reference information for the other sectors. The next sector, indexed as sector 0, contains the FAT. The FAT is simply an array of 32 bit integers, in where the array index represents the sector number and the value indicates the next sector in the chain. Sector chains relate data that is too large to fit into a single sector. Because the FAT resides in one or more sectors, a FAT directory structure (i.e., the DIFAT) exists to define the chained sectors for the FAT. For these structures, the value FAT_ENDOFCHAIN (0xFFFFFFFF) indicates the end of a sector chain.

The sector approach poses obvious space inefficiencies. To address these inefficiencies, the format provides a mini FAT (smaller 64 byte sectors) that stores smaller data chunks that, like other sectors, can chain together. The directory structure, like the other CFB components, mirrors the functionality of a file system directory. The CFB header provides pointers to directory sectors that delineate the client data location.

Table 3-1. High-Level Logical Overview of the CFB Format

CFB Header (512 | 4096 bytes)
FAT (variable)
Mini FAT (variable)
Directory (variable)
Data Sectors

Table 3-1 shows the *logical* layout of a compound file. The FAT may not follow the CFB Header physically. It also is not necessary for the Mini Fat to follow the FAT. The CFB header will give the sector locations for each component. For example, a client program may store the FAT at sector 0x2A and store the actual file data in preceding sectors.

[2] MS-DOS and FAT are registered trademarks of Microsoft Corporation

3.1.1 "Hermaphrodoc" Files

File inspection and sanitization program developers should know that it is possible to store files of different clients (e.g., Word and PowerPoint) in the same physical file. In this case, the client program will open and read its file content from the appropriate client stream and ignore the content of other client streams. If a user changes the file extension to the second client type, that client will read its data and ignore the original content. In all cases, data from the other client persists.

3.1.2 A Note about CFB Validation

File inspection and sanitization programs should load and verify all low-level CFB components prior to analyzing higher-level data. Details and suggestions on how to validate a CFB and its low-level components (e.g., FAT and DIFAT) appear in Section 7.

OFFICE 2003.3.1: CFB Header

DESCRIPTION:
The CFB header contains characteristic information about the file, including number of sectors, sector size, and directory layout. Clients use this information to understand how to process the remaining file components.

CONCERN:
Alteration of data structures within the CFB could lead to data hiding; specifically, in the DIFAT array. A user could hide data in the unused bytes of the DIFAT array, which occupies the last 360 bytes of the header. Office 2003 programs write OxFFFFFFFF (FAT_FRESECT) to the entry if no FAT sector exists for that given index. However, it is possible to write arbitrary data to that entry, which will remain undetected by Office 20003 client programs.

PRODUCT: WORD, EXCEL, POWERPOINT

LOCATION:
The CFB header is located at byte 0 of the Office 2003 document. If the file exists in memory, it must be aligned on a 64-bit boundary. The header occupies 512 bytes (total size). It consists of a signature, directory information, and other file data.

RECOMMENDATION:
AR.1 Validate: Ensure referential integrity in the CFB header and referenced sectors by checking the data members for consistency and verifying that all referenced data members are used in the file.

AR.2 Remove: N/A

AR.3 Replace: Replace data in unused data members or overwrite with 0xFF.

AR.4 External Filtering Required: N/A

AR.5 Review: N/A

EXAMPLE:

The following hexadecimal dumps show the content of a native CFH and the content of the same CFH after a utility program wrote a string to the DIFAT array. Word 2003 and Word 2007 read the corrupt file with no warnings.

```
0x00000000 D0 CF 11 E0 A1 B1 1A E1 00 00 00 00 00 00 00 00  ................
0x00000010 00 00 00 00 00 00 00 00 3E 00 03 00 FE FF 09 00  ........>.......
0x00000020 06 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00  ................
0x00000030 2A 00 00 00 00 00 00 00 00 10 00 00 2C 00 00 00  *...........,...
0x00000040 01 00 00 00 FE FF FF FF 00 00 00 00 29 00 00 00  ............)...
0x00000050 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x00000060 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x00000070 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x00000080 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x00000090 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x000000A0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x000000B0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x000000C0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x000000D0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x000000E0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x000000F0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x00000100 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x00000110 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x00000120 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x00000130 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x00000140 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x00000150 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x00000160 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x00000170 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x00000180 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x00000190 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x000001A0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x000001B0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x000001C0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x000001D0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x000001E0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x000001F0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
```

After running utility program:

```
0x00000000 D0 CF 11 E0 A1 B1 1A E1 00 00 00 00 00 00 00 00  ................
0x00000010 00 00 00 00 00 00 00 00 3E 00 03 00 FE FF 09 00  ........>.......
0x00000020 06 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00  ................
0x00000030 2A 00 00 00 00 00 00 00 00 10 00 00 2C 00 00 00  *...........,...
0x00000040 01 00 00 00 FE FF FF FF 00 00 00 00 29 00 00 00  ............)...
0x00000050 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x00000060 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x00000070 FF FF FF FF 54 68 69 73 20 69 73 20 74 6F 61 74  ....This is
0x00000080 61 6C 6C 79 20 62 6F 67 75 73 20 64 61 74 61 2E  bogus data.
0x00000090 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x000000A0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x000000B0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x000000C0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x000000D0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x000000E0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x000000F0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x00000100 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x00000110 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x00000120 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x00000130 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x00000140 FF FF FF FSF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x00000150 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x00000160 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x00000170 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
```

```
0x00000180 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x00000190 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x000001A0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x000001B0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x000001C0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x000001D0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x000001E0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0x000001F0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
```

REFERENCE:
See [MS-CFB], Section 2.2 Compound File Header.

OFFICE 2003.3.1: END

OFFICE 2003.3.2: Directory and Directory Entry

DESCRIPTION:

The Directory Entry serves as the reference for a specific entity within the file. For improved search performance, the directory is organized into a red-black tree and a self-balancing binary search tree. Each directory entry contains elements to support that algorithm.

CONCERN:

The directory serves as the map to the document. Modifying an entry could corrupt the document and make it unreadable. Modifying a directory entry also could cause client programs to ignore that entry, and potentially its contents, thereby providing a method for hiding data.

The directory data structure is a red-black tree, a type of binary search tree and algorithm that keeps the tree balanced. Invalid tree manipulation (i.e., changing data members that are outlined in the grammar) can result in a corrupt document. In particular, manipulation to hide a previously embedded OLE object will corrupt the parent document. Although a red-black data structure is used here, clients may not adhere strictly to the red-black tree algorithms.

PRODUCT: WORD, EXCEL, POWERPOINT

LOCATION:
The directory begins after the CFB Header, the FAT, and the mini FAT (near the start of the file). Its exact offset varies, depending on the FAT and mini FAT size. The directory simply is a variable- sized array of directory entries.

RECOMMENDATION:
AR.1 Validate: Ensure referential integrity and consistency in the directory by searching for entries to non-existing objects, objects lacking directory entries, unreferenced sectors, sectors not holding storage or streams, and data “fill” in the Mini FAT and FAT sectors.

AR.2 Remove: Remove directory entries that lack referential integrity by searching for entries that reference non-existent objects.

AR.3 Remove: Remove objects in the file that lacks a directory entry.

AR.3 Replace: Replace data "fill" in the Mini FAT and FAT by overwriting unused entries with 0xFF.

AR.4 External Filtering Required: N/A

AR.5 Review: N/A

REFERENCE:
See [MS-CFB], Section 2.6 Compound File Directory Sectors

OFFICE 2003.3.2: END

OFFICE 2003.3.3: Microsoft GUID

DESCRIPTION:
Microsoft implements the standards- based, Uniquely Universal Identifier (UUID), which it calls the Globally Unique Identifier (GUID). The GUID is a 128-bit number that has guaranteed uniqueness within a given context. The GUID is commonly broken into four integers of varying size. GUIDs are used liberally throughout the Office framework, the OLE/CFB architecture, and may other applications.

CONCERN:
The major issue with GUIDs occurred in earlier versions of the algorithm generating them. The algorithm embedded the host computer's Media Access Control (MAC) address and time stamp in the GUID.

If disclosure of the machine hardware address is a concern, file inspection and sanitization programs should determine what algorithm version generated an encountered GUID. To determine the version, programs can look at the most significant four bits of <Data3>. If version 1 generated the GUID, the GUID contains the MAC address of the computer that created the GUID and a time stamp of when it created the GUID. Because many inter-dependencies exist among the OLE components that use GUIDs, changing, removing, or zeroing out the GUID can cause operational issues within the OLE framework. File inspection and sanitization programs concerned with exposing the machine's hardware address may choose to change the MAC portion of the GUID, but must consider the uniqueness attribute that is crucial to the GUID and OLE framework.

PRODUCT: WORD, EXCEL, POWERPOINT

LOCATION:
N/A

RECOMMENDATION:

AR.1 Validate: Ensure the UUID and GUID are properly formatted.

AR.2 Remove: N/A

AR.3 Replace: Replace the GUID with a new GUID, if the original was created using version 1 of the UUID standard. Note that this may have operational impact on the underlying OLE and CFB frameworks.

AR.4 External Filtering Required: N/A

AR.5 Review: N/A

EXAMPLE:
The following script output shows a system hardware address and GUID that were generated using the version 1 algorithm:

```
taos $ ~/uuidtest.py

Hardware address: 08:00:27:53:e5:11

d7164542-9f1f-11de-861b-08002753e511

taos $ exit
```

Note that the last six bytes of the GUID match the hardware address from the previous line. Also, the first bit of the third field in the GUID, which corresponds to the first four bits of <data3>, is 1, indicating the version number.

REFERENCE:
See [RFC-4122].

See [MS-SECO], Section 2.5.5 GUIDs.

OFFICE 2003.3.3: END

3.2 Embedded and Linked Objects

The following discussion examines various methods that Microsoft uses to store external data. Embedded objects are data that are actually contained within the CFB file. Linked objects use data structures to reference objects that are external to the CFB (e.g., a file stored on a locally available file system).

At a high level, the logical layout of an OLE object with a CFB file looks like this:

Container App. Data	Creator App. Data	Embedded Object Native Data	Embedded Object Presentation Data

Figure 3-2. Logical Layout of Embedded OLE Object

Contained App. Data	Linked File Source Data	Linked Object Presentation Data

Figure 3-3. Logical Layout of Linked OLE Object

OFFICE 2003.3.4: Embedded Objects

DESCRIPTION:

As noted, the CFB format can store arbitrary data streams. The OLE format (protocol) defines how client programs can store and represent these data streams. Files with linked or embedded objects are referred to as "containers." Programs to create, edit, and view an object are called "creators."

Client programs can use either OLE 1.0 or OLE 2.0 formats to store the actual data stream. In either case, the data stream (storage) is layered on top of the CFB format, which is also considered an OLE object.

For embedded objects, the creator application data and the embedded object's native data can be CFB files. This is the case for embedded spreadsheets within a word processing document.

Note: Two universal recommendations for constructs that deal with presentation objects include the following: (1) ensure that the presentation data matches data it represents, and (2) ensure that every linked or embedded object has an associated presentation object. These recommendations, although redundant, repeat for each construct to ensure completeness.

Also, file type associations that reside in the files may not be accurate (e.g., a Word file believes that an embedded object is an image but the attached file contains a program). Therefore, additional verification using the content of embedded files may be warranted.

CONCERN:

Embedded objects carry all security concerns of the involved file format. Consider them as data disclosure and attack concerns.

RECOMMENDATION:

AR.1 Validate: Verify the OLE structure and that the embedded object is appropriate for the type of data stream.

AR.2 Remove: Remove all embedded objects that the document does not reference explicitly.

AR.3 Remove: Remove both the object and any references to the object.

AR.4 Replace: N/A

AR.5 External Filtering Required: Pass the embedded object to the action configured for its data type.

AR.6 Review: Present the object for review.

OFFICE 2003.3.4: END

3.2.1 OLE 1.0 Constructs

One may see the following constructs (OFFICE 2003.3.5: to OFFICE 2003.3.13:) rarely in Office 2003 formatted files; however, we included them for completeness.

OFFICE 2003.3.5: Presentation Object Header

DESCRIPTION:
This construct precedes the presentation data - the information about the contained object that the container requires to properly display the linked or embedded object.

CONCERN:
Modification of the presentation object header could alter the way a container presents the object - perhaps even hiding the object.

PRODUCT: WORD, EXCEL, POWERPOINT

LOCATION:
The actual location is within the data stream that contains the object (prior to presentation data).

RECOMMENDATION:
AR.1 Validate: Ensure referential integrity and consistency in the header and that the header object is appropriate for the object type.

AR.2 Remove: Remove all headers that lack referential integrity: headers that the document does not reference explicitly and headers for non-existent objects.

AR.3 Replace: Restore referential integrity, if possible, by reinserting missing objects that a header referenced.

AR.4 External Filtering Required: N/A

AR.5 Review: N/A

REFERENCE:
See [MS-OLEDS], Section 2.2.1 PresentationObjectHeader

OFFICE 2003.3.5: END

OFFICE 2003.3.6: Standard Presentation Object

DESCRIPTION:
This data structure contains data for presenting an OLE 1.0 object.

CONCERN:
The presentation object tells the container object how to present the object.

PRODUCT: WORD, EXCEL, POWERPOINT

LOCATION:
The presentation data logically follows the native data in the OLE 1.0 data stream.

RECOMMENDATION:

AR.1 Validate: Ensure referential integrity and consistency by verifying that the presentation header properly references the standard presentation object and by verifying the existence and correct type of the target of the standard presentation object.

AR.2 Remove: Remove all objects that the document does not reference explicitly.

AR.3 Remove: Remove the object and any references to the object.

AR.4 Replace: N/A

AR.5 External Filtering Required: Pass the object and the object native data type to the action that is configured for applicable data types.

AR.6 Review: Present the object for review.

REFERENCE:
See [MS-OLEDS], Section, 2.2.2 StandardPresentationObject

OFFICE 2003.3.6: END

OFFICE 2003.3.7: Metafile Presentation Object

DESCRIPTION:
This object describes the presentation information for metafile objects.

CONCERN:
All presentation objects should be present when embedded or linked objects are contained within a CFB.

PRODUCT: WORD, EXCEL, POWERPOINT

LOCATION:
This object follows the native data stream.

RECOMMENDATION:

AR.1 Validate: Ensure referential integrity and consistency by verifying that the presentation header properly references the metafile presentation object and by verifying the existence and correct type of the target of the metafile presentation object.

AR.2 Remove: Remove all objects that the document does not reference explicitly.

AR.3 Remove: Remove the object and any references to the object.

AR.4 Replace: N/A

AR.5 External Filtering Required: Pass the object and the object native data type to the action that is configured for applicable data types.

AR.6 Review: Present the object for review.

REFERENCE:

See [MS-OLEDS], Section 2.2.2.1 MetaFilePresentationObject

OFFICE 2003.3.7: END

OFFICE 2003.3.8: Bitmap Presentation Object

DESCRIPTION:

This object describes the presentation information for bitmap objects.

CONCERN:

All presentation objects should be present when embedded or linked objects are contained within a CFB.

PRODUCT: WORD, EXCEL, POWERPOINT

LOCATION:

This object follows the native data stream.

RECOMMENDATION:

AR.1 Validate: Ensure referential integrity and consistency by verifying that the presentation header properly references the bitmap presentation object and by verifying the existence and correct type of the target of the bitmap presentation object.

AR.2 Remove: Remove all objects that the document does not reference explicitly.

AR.3 Remove: Remove the object and any references to the object.

AR.4 Replace: N/A

AR.5 External Filtering Required: Pass the object and the object native data type to the action that is configured for applicable data types.

AR.6 Review: Present the object for review.

REFERENCE:

See [MS-OLEDS], Section 2.2.2.2 BitmapPresentationObject

OFFICE 2003.3.8: END

OFFICE 2003.3.9: DIB Presentation Object

DESCRIPTION:

This object describes the presentation information for Device-Independent Bitmap (DIB) objects.

CONCERN:
All presentation objects should be present when embedded or linked objects are contained within a CFB.

PRODUCT: WORD, EXCEL, POWERPOINT

LOCATION:
This object follows the native data stream.

RECOMMENDATION:

AR.1 Validate: Ensure referential integrity and consistency by verifying that the presentation header properly references the DIB presentation object and by verifying the existence and correct type of the target of the DIB presentation object.

AR.2 Remove: Remove all objects that the document does not reference explicitly.

AR.3 Remove: Remove the object and any references to the object.

AR.4 Replace: N/A

AR.5 External Filtering Required: Pass the object and the object native data type to the action that is configured for applicable data types.

AR.6 Review: Present the object for review.

REFERENCE:
See [MS-OLEDS], Section 2.2.2.3 DIBPresentationObject

OFFICE 2003.3.9: END

OFFICE 2003.3.10: Generic Presentation Object

DESCRIPTION:
This object describes the format for clipboard data. There are two types of generic presentation objects: standard clipboard format and registered clipboard format.

CONCERN:
All presentation objects should be present when embedded or linked objects are contained within a CFB.

PRODUCT: WORD, EXCEL, POWERPOINT

LOCATION:
This object follows the native data stream.

RECOMMENDATION:

AR.1 Validate: Ensure referential integrity and consistency by verifying that the presentation header properly references the generic presentation object and by verifying the existence and

correct type of the target of the generic presentation object.

AR.2 Remove: Remove all objects that the document does not reference explicitly.

AR.3 Remove: Remove the object and any references to the object.

AR.4 Replace: N/A

AR.5 External Filtering Required: Pass the object and the object native data type to the action that is configured for the applicable data type.

AR.6 Review: Present the object for review.

REFERENCE:
See [MS-OLEDS], Section 2.2.3.3 RegisteredClipboardFormatPresentationObject

OFFICE 2003.3.10: END

OFFICE 2003.3.11: OLE 1.0 Object Header

DESCRIPTION:
This data structure precedes any linked or embedded OLE 1.0 object.

CONCERN:
As with other header information, inconsistency in the header may indicate an attempt to hide or misrepresent data within the CFB.

PRODUCT: WORD, EXCEL, POWERPOINT

LOCATION:
The object header resides at the beginning of an OLE 1.0 data stream - usually at the start of a CFB sector.

RECOMMENDATION:

AR.1 Validate: Ensure referential integrity and consistency of the header by verifying that the target object exists and the header's data members correctly identify the target object.

AR.2 Remove: Remove the header if the target object is removed.

AR.3 Replace: N/A

AR.4 External Filtering Required: Pass the header to the configured application for its data type if the target object has been passed for external filtering.

AR.5 Review: N/A

REFERENCE:
See [MS-OLEDS], Section 2.2.4 ObjectHeader

OFFICE 2003.3.11: END

OFFICE 2003.3.12: OLE 1.0 Embedded Object

DESCRIPTION:
This data structure holds the data of an embedded object.

CONCERN:
All embedded objects can contain the same potential flaws as the container document and should undergo the same file inspection and sanitization procedures as the root document.

PRODUCT: WORD, EXCEL, POWERPOINT

LOCATION:
Embedded objects follow the object header.

RECOMMENDATION:

AR.1 Validate: Ensure referential integrity with the applicable presentation object and object header.

AR.2 Remove: Remove the data structure and the embedded data.

AR.3 Replace: N/A

AR.4 External Filtering Required: Pass the native data to the action that configured its data type.

AR.5 Review: N/A

REFERENCE:

See [MS-OLEDS], Section 2.2.4 ObjectHeader

OFFICE 2003.3.12: END

OFFICE 2003.3.13: OLE 1.0 Linked Object

DESCRIPTION:
This data structure holds the data of linked object.

CONCERN:
As with embedded objects, linked objects have the same flaws and areas of concern as the root document. However, because linked object data does not reside physically in the data store of the root CFB, file inspection and sanitization programs may choose to handle linked objects differently than embedded objects.

Linked object data structures within a file can point to external resources on a secure network.

Leaving a linked object's data structures in place can lead to inadvertent data leakage (e.g., of directory structures, network addresses and names and other sensitive data).

PRODUCT: WORD, EXCEL, POWERPOINT

LOCATION:
Linked objects follow the object header.

RECOMMENDATION:

AR.1 Validate: Ensure the internal referential integrity of the object, its presentation data, and the object header. Ensure the external referential integrity of the object by verifying the target object's existence and that accessibility of this resource to users following the data transfer.

AR.2 Remove: Remove all data structures that are associated with the linked object.

AR.3 Remove: If the linked object resource is not available to users following the data transfer, remove the linked object and its associated data structures.

AR.4 Replace: N/A

AR.5 External Filtering Required: If the linked object resource is accessible to users following the data transfer, pass the resource to the action that configured its data type.

AR.6 Review: N/A

REFERENCE:
See [MS-OLEDS], Section 2.2.4 ObjectHeader

OFFICE 2003.3.13: END

3.2.2 OLE 2.0 Constructs

OFFICE 2003.3.14: ClipboardFormatorAnsiString

DESCRIPTION:
This data structure stores either a numerical identifier for a standard clipboard format or an American National Standards Institute (ANSI) string that denotes the standard clipboard format. Client programs use the first field as a tag to denote the type of data stored in the object.

CONCERN:
Incorrect manipulation of this field could lead to data hiding or prevent users from properly accessing an OLE data stream.

PRODUCT: WORD, EXCEL, POWERPOINT

RECOMMENDATION:

AR.1 Validate: Ensure internal consistency in the data structure by using the *MarkerOrLength* field to determine the type of data stored in the object, verifying the data is either a numeric value

representing a standard clipboard format or a null-terminated ANSI string with length equal to *MarkedOrLength* (including the null termination), and verifying that the string represents the name of a registered clipboard format.

AR.2 Remove: N/A

AR.3 Replace: N/A

AR.4 External Filtering Required: If the object contains an ANSI string, pass the string to the action that is configured for text objects.

AR.5 Review: N/A

REFERENCE:
See [MS-OLEDS], Section 2.3.1 ClipboardFormatOrAnsiString

OFFICE 2003.3.14: END

OFFICE 2003.3.15: ClipboardFormatOrUnicodeString

DESCRIPTION:
This data structure stores either a numerical identifier for a standard clipboard format or a Unicode string that denotes the standard clipboard format. Client programs use the first field as a tag to denote the type of data stored in the object.

CONCERN:
Incorrect manipulation of this field could lead to data hiding or prevent users from properly accessing an OLE data stream.

PRODUCT: WORD, EXCEL, POWERPOINT

RECOMMENDATION:

AR.1 Validate: Ensure internal consistency in the data structure by using the *MarkerOrLength* field to determine the type of data stored in the object, verifying the data is either a numeric value representing a standard clipboard format or a null-terminated Unicode string with length equal to *MarkedOrLength* (including the null termination), and verifying that the string represents the name of a registered clipboard format.

AR.2 Remove: N/A

AR.3 Replace: N/A

AR.4 External Filtering Required: If the object contains a Unicode string, pass the string to the action that is configured for text objects.

AR.5 Review: N/A

REFERENCE:

See [MS-OLEDS], Sections 2.3.2, ClipboardFormatOrUnicodeString

OFFICE 2003.3.15: END

OFFICE 2003.3.16: OLE 2.0 Stream (OLEStream)

DESCRIPTION:

Client programs use this data stream to store arbitrary OLE objects.

CONCERN:

As with the OLE 1.0 data objects, the actual data stream within an OLE object can harbor the same flaws as the root document, such as data hiding and exploitable software flaws.

PRODUCT: WORD, EXCEL, POWERPOINT

LOCATION:

OLE Stream structures are located in the OLE Compound File Stream of the CFB, where a name of the stream is "\1OLE". The stream object itself (for both linked data and embedded data) resides in the OLE Compound File Storage.

RECOMMENDATION:

AR.1 Validate: Ensure the consistency of the OLEStream by determining if the *Version* and *Flags* attributes are set correctly and using these settings to verify correctness of the other data structure attributes. Ensure the referential integrity for embedded objects and internal referential integrity for linked objects by verifying the references to the object in the OLEStream structure. Ensure the external referential integrity of a linked object by verifying the existence of the target.

AR.2 Remove: Remove the OLEStream by overwriting location fields in the structure with appropriate values and writing either 0x00000000 or 0xFFFFFFF to the storage.

AR.3 Replace: N/A

AR.4 External Filtering Required: Pass the native data in the storage to the action that is configured for the applicable data type.

AR.5 Review: N/A

REFERENCE:

See [MS-OLEDS], Sections 2.3.3 OLEStream and 2.3.3.1 MONIKERSTREAM

OFFICE 2003.3.16: END

OFFICE 2003.3.17: OLE 2.0 Presentation Stream

DESCRIPTION:

The OLE 2.0 Presentation Stream controls the presentation data for the containing OLE object. At most 999 streams can exist. In the CFB file, the streams are named "\20OlePresxxx," where x represents a digit.

CONCERN:

Invalid presentation data can represent an attempt to alter or hide OLE data.

PRODUCT: WORD, EXCEL, POWERPOINT

LOCATION:

The OLE presentation streams start on sector boundaries within the CFB. The CFB directory contains pointer information for the actual sector location.

RECOMMENDATION:

AR.1 Validate: Ensure referential integrity by verifying that the presentation header information matches that of the target object. Ensure consistency in the presentation stream by scanning the header to ensure that the OLE 2.0 presentation stream matches the OLE object it represents

AR.2 Remove: Remove the presentation object, if the target object is removed, by deleting the presentation object from the containing stream. (Because the presentation object resides in the object storage, remove the object storage.)

AR.3 Replace: N/A

AR.4 External Filtering Required: Identify the type of data and submit the data to the action that is configured for the data type.

AR.5 Review: N/A

REFERENCE:

See [MS-OLEDS], Section 2.3.4 OLEPresentationStream

OFFICE 2003.3.17: END

OFFICE 2003.3.18: OLE 2.0 Native Data Stream (OLENativeStream

DESCRIPTION:

The CFB stores the native data of the OLE 1.0 object in a native data stream.

CONCERN:

The OLE object contains arbitrary data of an arbitrary format; therefore, it is subject to the same concerns as that format. Some researchers have noted that the length field in this stream does not

match the actual stream length

PRODUCT: WORD, EXCEL, POWERPOINT

LOCATION:
The native data stream resides within the OLE Compound File Stream named "\1Ole10Native" within the Compound File Storage that corresponds to this object.

RECOMMENDATION:

AR.1 Validate: Ensure the consistency of the native data stream by checking the value of the *NativeDataSize* attribute and verifying that the native data is of that size.

AR.2 Remove: N/A

AR.3 Replace: N/A

AR.4 External Filtering Required: Identify the data type and submit the data to the action that is configured for that data type.

AR.5 Review: N/A

REFERENCE:
See [MS-OLEDS], Section 2.3.6 OLENativeStream

OFFICE 2003.3.18: END

OFFICE 2003.3.19: Compound Object Header

DESCRIPTION:
This data structure stores information that is related to the associated compound object stream.

CONCERN:
Apparently, the data structure is a placeholder. Microsoft's documentation states that data can be of an arbitrary value and must be ignored. Therefore, a potential for data hiding in this structure exists.

PRODUCT: WORD, EXCEL, POWERPOINT

LOCATION:
The compound object header is in the first 28 bytes of the compound object stream.

RECOMMENDATION:

AR.1 Validate: N/A

AR.2 Remove: N/A

AR.3 Replace: Replace values in the data structure with 0x00000000 or 0xFFFFFFFF.

AR.4 External Filtering Required: N/A

AR.5 Review: N/A

REFERENCE:
See [MS-OLEDS], Section 2.3.7 CompObjHeader

OFFICE 2003.3.19: END

OFFICE 2003.3.20: Component Object Stream

DESCRIPTION:
The compound object stream contains information about the clipboard format and presentation data for the associated OLE object. The name of the stream is "\1CompObj."

CONCERN:
Incorrect presentation data may represent an attempt to hide or alter data.

PRODUCT: WORD

LOCATION:
The Compound Object Stream is located within OLE Compound file stream for the associated object.

RECOMMENDATION:

AR.1 Validate: Ensure consistency by verifying that the attributes following the 28 byte header correctly identify the clipboard format and presentation data for the object.

AR.2 Remove: N/A

AR.3 Replace: N/A

AR.4 External Filtering Required: Identify the type of the presentation data and pass the data to the action that is configured for that data type.

AR.5 Review: N/A

REFERENCE:
See [MS-OLEDS], Section 2.3.8 CompObjStream

OFFICE 2003.3.20: END

3.2.3 General Notes Regarding OLE

This section provides noteworthy information about recursive object processing, masquerading objects, and foreign data types.

3.2.3.1 Recursive Object Processing

As noted previously, MS Office uses the OLE standard to store arbitrary data within Office 2003 documents. Arbitrary data may include other Office documents, multimedia files, and binary data (including executable images).

File inspection and sanitization programs must examine these OLE objects recursively, treating the objects as they would treat the base document. The design of these programs lies beyond the document's scope; however, architects and software engineers should ensure their programs recursively process embedded objects and consider the results from recursive processing as part of the results of the enclosing document.

For example, if a Word document contains an Excel file, the inspection or sanitization program should process the Excel file as if the Excel file was passed to it originally. The application processing the Excel file should report its results to the original program. Processing order is a design issue. In some cases, a program may choose to perform the recursive analysis inline or after processing the rest of the original document. Likewise, it may choose to delegate the processing to a new thread or process. In all cases, one should not consider the inspection or sanitization of a base document complete until after inspecting or sanitizing all contained OLE objects.

3.2.3.2 Masquerading Objects

Using the "Insert Object" feature of MS Office products, users easily can insert arbitrary files, including executable images, into their documents. The client programs store these objects in the OLE data structures that are summarized above. The client programs also will warn users if they subsequently attempt to access the object; however, it is easy to convert the icon representing the object to something innocuous (such as a file folder icon) that does not truly represent the underlying object.

A possible solution to this issue is to examine embedded objects to ensure the document represented the object correctly. For example, the following data snippet from a Word document shows an object as a *JPEG* image when, in fact, the object is the *ping* executable image. (The two bytes highlighted in red are the executable file signature.)

```
00003000: 0270 0000 0200 426c 7565 2068 696c 6c73  .p....Blue hills
00003010: 2e6a 7067 0043 3a5c 444f 4355 4d45 7e31  .jpg.C:\DOCUME~1
```

```
00003020: 5c41 4c4c 5553 457e 315c 444f 4355 4d45  \ALLUSE~1\DOCUME
00003030: 7e31 5c4d 5950 4943 547e 315c 5341 4d50  ~1\MYPICT~1\SAMP
00003040: 4c45 7e31 5c42 4c55 4548 497e 312e 4a50  LE~1\BLUEHI~1.JP
00003050: 4700 0000 0300 3d00 0000 433a 5c44 4f43  G.....=...C:\DOC
00003060: 554d 457e 315c 414c 4c55 5345 7e31 5c44  UME~1\ALLUSE~1\D
00003070: 4f43 554d 457e 315c 4d59 5049 4354 7e31  OCUME~1\MYPICT~1
00003080: 5c53 414d 504c 457e 315c 424c 5545 4849  \SAMPLE~1\BLUEHI
00003090: 7e31 2e4a 5047 0069 6f4d 5a90 0003 0000  ~1.JPG.ioMZ.....
000030a0: 0004 0000 00ff ff00 00b8 0000 0000 0000  ................
000030b0: 0040 0000 0000 0000 0000 0000 0000 0000  .@..............
000030c0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000030d0: 0000 0000 00d8 0000 000e 1fba 0e00 b409  ................
000030e0: cd21 b801 4ccd 2154 6869 7320 7072 6f67  .!..L.!This prog
000030f0: 7261 6d20 6361 6e6e 6f74 2062 6520 7275  ram cannot be ru
00003100: 6e20 696e 2044 4f53 206d 6f64 652e 0d0d  n in DOS mode...
00003110: 0a24 0000 0000 0000 0059 96f6 681d f798  .$.......Y..h...
00003120: 3b1d f798 3b1d f798 3bde f897 3b1c f798  ;...;...;...;...
```

3.2.3.3 Foreign Data Types

In addition to embedded files (e.g., an Excel file within a Word document), the OLE format also supports embedding foreign data types, such as an Excel chart in Word. Sometimes the client program simply converts the data type to a native format (e.g., Word converts copied Excel cells into a native table). Other times, the object becomes represented as other OLE embedded types are.

For the latter, as with OLE files, file inspection and sanitization programs could process the OLE object in the appropriate manner for that data type. In most instances in which the client program converts the data to its native format, the file inspection and sanitization program can process the data as part of the enclosing document.

3.3 Embedded Fonts in Office Documents

To ensure correct presentation of typography in an Office document, users can embed TrueType or Open Type fonts. When processing a document with embedded fonts, client applications will install the font temporarily if it is not available on the target computer. This feature has been a source of problems for the Office product suite and web applications (for example, refer to this Microsoft security bulletin).

The Worley Wide Web Consortium defines a standard for embedded fonts, based in large part on Microsoft's work in the area.

OFFICE 2003.3.21: Embedded Font Structures

DESCRIPTION:

An embedded font structure holds the information needed to temporarily install a font for rendering on a target system. Three versions of the structure are currently in use; their current structure is documented at W3C Embedded Fonts.

The structures can hold arbitrary data of varied size.

CONCERN:

Because the structures can hold arbitrary data of varied size, they can be used easily for both data hiding and exploitation of software flaws within client applications. As noted, this data structure has been used for such activities.

PRODUCT: WORD

LOCATION:

The embedded font structures are indexed in a sub-type of the FIB, the *FibRgFcLcb97*. Within this very large data structure are two data members of interest: *fcSttbTtmbd* and *lcbSttbTtmbd*. *fcSttbTtmbd* represents an offset into the *TableStream* and *lcbSttbTtmbd* indicates the size of the *sttrTtmbd* that is stored at *fcSttbTtmbd*. A *strrTttmbd* stores a list of *Tttmd* structures, which contain information about the font, enclosing the location of the actual font in the WordDocument stream

PRODUCT: POWERPOINT

LOCATION:

The location of embedded fonts in PowerPoint follows from the DocumentContainer. The DocumentContainer contains a data member called *documentTestInfo*, a *docmentTextInfoCotainer* record. That record contains a member called *fontCollection*, which is a *fontCollectionContainer*. Within that record is a date member called the *rgFontCollectionEntity*, an array of *FontCollectionEntity* objects. A *FontCollectionEntity* contains four variable sized members containing the actual embedded font data.

RECOMMENDATION:

WR.1 Validate: Ensure consistency by verifying the values stored in f*cSttbTtmbd* and *lcbSttbTtmbd*. Ensure referential integrity and data integrity by examining the *Tttmd* data members and verifying that those attributes point to a valid, embedded font.

PR.1 Validate: Ensure consistency, referential integrity and data integrity by parsing and examining the DocumentContainer data structure hierarchy and verifying that data in each *FontCollectionEntry* contains valid embedded fonts.

AR.2 Remove: Remove the embedded font by zeroing out its location within the appropriate storage and adjusting the appropriate data structure members that are outlined in the Location sections above.

AR.3 Replace: N/A

AR.4 External Filtering Required: Present the data structure members for the embedded fonts to the filter for fonts.

AR.5 Review: N/A

REFERENCE:

See [MS-DOC], Section 2.9.328, Ttmd; Section 2.5.6, FibRgFcLcb97.

See [MS-PPT], Section 2.9, Text Types.

OFFICE 2003.3.21: END

3.4 MS Word-specific File Level Constructs

Layered on top of the OLE CFB are Word-specific constructs that define the data stream for a word processing document. This section examines the details of those constructs.

A Word document breaks data into objects referred to as *streams* or *storage*. The main data stream within a Word file is the *WordDocument* stream. This stream commonly begins immediately after the CFB header and starts at an offset of 0, relative to the end of the CFB header sector, either at raw byte 512 (0x200) or 4096 (0x1000), depending on the sector size. However, developers must always reference the CFB directory structure to determine the precise location. The *WordDocument* stream must always begin with a FIB, which defines the underlying data locations for the stream.

The document may also contain the following:

- A variant of a table stream
- A data stream
- ObjectPool storage (if OLE objects are embedded in the document)
- A custom XML data stream (optional)

  Must be named “MsoDataStore”

- A summary information data stream (optional)

  Must be named “\005Summary Information,” where \005 represents the hex value 0x0005

- A document summary information stream (optional)

  Must be named “\005Document Summary Information,” where \005 represents the hex value of 0x0005

- An encryption stream (optional)

Must be named “encryption”

- Macro storage (optional)

  Must be a Property Root Storage

- XML signature storage (optional)

  Must be named “_xmlsignature”

- Signature storage (optional)

  Must be named “_signature”

- Information Rights Management data space storage (optional)
  - Must be named “\006DataSpaces,” where \006 is hex value 0x0005
  - If present, protected content storage must also be present

- Protected content stream (optional)

  Must be named “\009DRMContent,” where \009 is hex value 0x0009

The presence of the Information Rights Management data stream and the Protect Content stream forces certain behaviors on clients. Specifically, the client must ignore all other (non-protected) storages and streams. However, this provides a mechanism for data hiding.

This document will not cover all of the possible Word-specific constructs. Readers should consult MS-DOC.pdf for detailed information on the low-level constructs (e.g., for formatting and page layout). The data constructs that appear below are representative examples of Word data structures.

OFFICE 2003.3.22: FIB

DESCRIPTION:
The FIB anchors the entire document and provides pointers to the different components of the

document. This complex structure of variable size consists of several variable length sub-structures that Word uses internally.

Every Word version (e.g., Word 97, Word XP) added FIB functionality to the FIB and extended the data structure to support this new functionality. Like the base FIB, these data structures are large (variable size) and complex. Therefore, full treatment of these data structures lies beyond the scope of this document. Readers should consult the official Microsoft documentation.

CONCERN:
Because the FIB provides the roadmap to data within the file, altering its contents can have an adverse effect on the file, rendering the document or parts of the document unreadable.

PRODUCT: WORD

LOCATION:
The FIB is usually located at byte 0 relative to the end of the OLE CFB header, which is either byte 512 or 4086, depending on sector size. The byte position of the *WordDocument* stream within the CFB directory determines the exact location.

RECOMMENDATION:

WR.1 Validate: Ensure consistency by checking that fields within the FIB, and its related or derived data structures, marked as "MUST BE" or "SHOULD BE" a certain value, are set to that value.

WR.2 Remove: Set fields within the FIB, and its related and derived data structures that are marked as "MUST BE IGNORED", to an appropriate value (e.g., 0x00 or 0xFF).

WR.3 Replace: N/A

WR.4 External Filtering Required: N/A

WR.5 Review: N/A

REFERENCE:

See [MS-DOC], Section 2.5.1 Fib

OFFICE 2003.3.22: END

3.5 MS Excel-specific File Level Constructs

Like MS Word, Excel layers its data on top of the CFB. For Excel, the basic data store is the record, a variable length structure consisting of a 16- bit integer tag, a 16 -bit integer record size, and a data store. A record cannot exceed 8224 bytes total length. To accommodate future applications, Excel supports a "future" record concept. Records generally are either a supported record or a future record. Future records act as

wrappers around record data. When encountering future records, applications can choose to ignore the "wrapped" data.

The future record architecture poses a unique challenge to file inspection and sanitization programs. By design, Excel will ignore unrecognized future records; however, it will persist these records when users store the file. This provides an ideal vector for hiding and transmitting data without user awareness.

MS Excel uses 355 different record types. It falls beyond the scope of this document to describe each in detail. Refer to the following Uniform Resource Locator (URL) for detailed descriptions:
https://msdn.microsoft.com/en-us/library/cc313154.aspx.

Note that for some records, data resides in the Binary Interchange File Format (BIFF).

3.5.1 Records

Excel supports more than 300 record types. For illustrative purposes, this document describes two record types: the general record type and the beginning of a file record.

OFFICE 2003.3.23: General Excel Record

DESCRIPTION:
Excel stores all data in records that are overlaid on the CFB sectors. A record can be up to 8224 bytes in size, greater than either sector size of 512 or 4096 bytes. Record data can also exceed 8224 bytes, in which case Excel uses "continue" records to store the excess data.

The record format is relatively straightforward: a tag, a size, and then the data.

CONCERN:
Areas of concern are minimal for the general Excel record; however, record sub-types may contain potential security flaws.

PRODUCT: EXCEL

LOCATION:
The general record data precedes any specific record information.

RECOMMENDATION
AR.1 Validate: N/A

AR.2 Remove: N/A

AR.3 Replace: N/A

AR.4 External Filtering Required: Present the text to the filter that is configured for text

AR.5 Review: N/A

EXAMPLE:

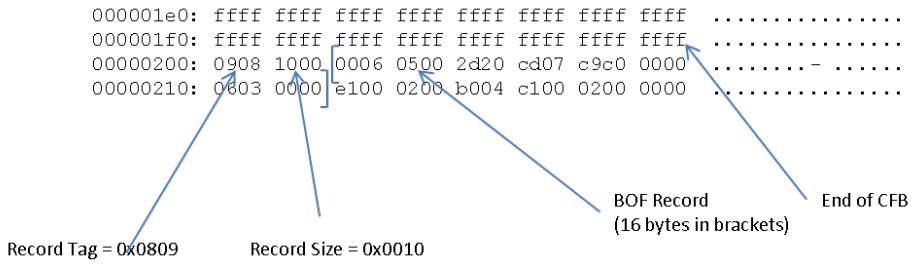

Figure 3-4. Excel Record, Showing General and Specific Record Data

REFERENCE:
See [MS-XLS], Section 2.4 Records

OFFICE 2003.3.23: END

OFFICE 2003.3.24: Beginning of File (BOF) Record

DESCRIPTION:
The BOF record starts an individual sub-stream that the workbook specifies.

CONCERN:
The record does not allow for variable sized data storage and would not be considered a good vector for data hiding; however, altering any of the settings within the record could have an adverse impact on the user's ability to read or use the file.

PRODUCT: EXCEL

LOCATION:
The location will vary, but every sub-stream begins with a BOF record.

RECOMMENDATION:
ER.1 Validate: Ensure consistency by verifying that the document type field is set to a proper value and that all fields marked as "MUST BE IGNORED" should be set to an appropriate value (e.g., 0x00 or 0xFF).

ER.2 Remove: N/A

ER.3 Replace: N/A

ER.4 External Filtering Required: N/A

ER.5 Review: N/A

EXAMPLE:
See Figure 3-4.

REFERENCE:
See [MS-XLS], Section 2.4.21 BOF

OFFICE 2003.3.24: END

3.5.2 Future Records

Future records enable client programs to extend the Excel record architecture by marking certain records as part of the future record architecture. Client programs that do not recognize a future record can ignore the record, but should preserve it when saving the overall file.

In the Excel 97-2003 format, future records can appear in either the Chart sheet sub-stream or the PivotTable sub-stream.

OFFICE 2003.3.25: Future Records

DESCRIPTION:
The future record architecture defines five future record headers: FRTHeader, FrtHeaderOld, FrtRefHeader, FrtRefHeaderNoGrBit, and or FrtRefHeaderU. Future record headers are contained within a record type.

CONCERN:
Future records provide a vector for persistent data hiding.

PRODUCT: EXCEL

LOCATION:
Future records almost uniformly reside in the first bytes immediately following the generic record information. Within a chart sheet sub-stream, future records must fall into one of three categories:

1) Chart-specific future records.

The record refers to a record with a type enumeration between 2048 and 2303, inclusive.

The record is contained in a collection that is either of the following:

The record is contained in a record collection marked by StartBlock and EndBlock records AND that collection itself is not part of another collection.

OR

The record is contained in a collection specified by the StartObject and EndObject records.

ChartFrtInfo must precede the first Chart-specific future record in the chart sheet sub-stream.

2) Storage of non-future records.

Non-future records are wrapped in the FRTWrapper.

The entire record must be stored within in the wrapperRecord field of the FRWrapper.

The records MUST exist in the collection specified by the StartObject and EndObject records.

3) A catch-all category of all other future records not belonging to (1) or (2) above.

Excel stores future records for the PivotTable sub-stream using the SXAddl record. [MS-XLS], Section 2.2.5.1.1 fully describes this.

RECOMMENDATION:

ER.1 Validate: Ensure consistency by scanning any future records and verifying that they conform to the future record standard.

ER.2 Remove: Remove unrecognized future records by overwriting them with 0x00 or 0xFF.

ER.3 Replace: N/A

ER.4 External Filtering Required: N/A

ER.5 Review: N/A

REFERENCE:
See [MS-XLS], Section 2.1.6 Future Record

OFFICE 2003.3.25: END

3.6 MS PowerPoint-specific File Level Constructs

Like Word and Excel, PowerPoint overlays its constructs on the OLE CFB-formatted data stream. In a PowerPoint document, objects, called atoms (records), are grouped together in collections. PowerPoint extensively uses CRB streams. Each one stores collections of records that hold user data and document metadata, as follows:

- Current User Stream: a required stream whose name must be "Current User" and holds the CurrentUserAtom record.
- PowerPointDocument Stream: the main stream that holds the document contents. It can contain any of the following collections of records, which are records themselves:
    - DocumentContainer
    - MasterOrSlideContainer

  - HandoutContainer
  - SlideContainer
  - NotesContainer
  - ExOleObjStg
  - ExControlStg
  - VbaProjectStg
  - PersistDirectoryAtom
  - UserEditAtom
  - Picture Stream

- Summary Information Stream: a required stream whose name must be "\005Summary Information," where "\005" represents 0x0005.
- Document Summary Information Stream: a stream whose name must be "\005DocumentSummaryInformation." This stream may be omitted if the file is encrypted.
- Encrypted Summary Information: a stream that exists for encrypted documents. It must be named "EncryptedSummary."
- Digital Signature Storage: a storage holding the digital signature. It must be named "_xmlsig." It may be omitted and ignored.
- Custom XML Storage: a storage holding custom XML. It must be named "MsoDataStore."
- Signature Stream: a stream that should be omitted and ignored. It must be named "_signature."

4. OFFICE 2003 CONSTRUCTS AND METADATA

OFFICE 2003.4.1: Field Codes

DESCRIPTION:

Field codes are used to automatically update certain information in an Office document. For example, a date field code will automatically update the text within that field code to the current date. MS Word has 75 different fields. Fields apply only to Word (i.e., they do not apply to PowerPoint or Excel). To view Fields in MS Word, open the application and select the pull-down Insert → Field. This displays the list of field items that a Word document can include.

Fields are included in Word's "smart formatting" feature. Fields have a text representation in the document. A field can also have a definition, which enables the field to update itself when selecting the Update Field option.

CONCERN:

Field codes could hide or obtain data from users without their knowledge, as fields may update each time a document opens. Fields also may include whole files and pictures from outside the document. This usage may reveal sensitive network architectures and could render parts of the document unreadable or missing after the document transfers outside of the domain in which the linked file exists.

Table 4-1. Field Codes

Category	Fields	Description & Concern
Date and Time	CREATEDATE DATE EDITTIME PRINTDATE SAVEDATE TIME	Inserts the current date and/or time, or date and/or time of some kind of event. Some dates can be important to the document functionality, but also may have the potential to leak information. CreateDate in a legal document can be vital information for document functionality. Date can be part of the basic document functionality, or it may reveal more information than desired by your site.
Document Automation	COMPARE DOCVARIABLE GOTOBUTTON IF MACROBUTTON PRINT	Compares values and takes action based on outcome, runs macros, and sends a code to a printer. If the document needs such automation, it is probably the main reason the document exists. Document automation also is associated with using Visual Basic in the document. Preserving document automation may also require leaving macros in the document. Obviously preserving document automation is risky, and getting such

		documents through a filter may require establishing a special policy for these files only.
Document Information	AUTHOR COMMENTS DOCPROPERTY FILENAME FILESIZE INFO KEYWORDS LASTSAVEDBY NUMCHARS NUMPAGES NUMWORDS SUBJECT TEMPLATE TITLE	Inserts or stores information about the document. A core set of basic fields provides support for nearly all Word documents. These fields are keys to preserving the ability to edit documents and present little threat.
Equations and Formulas	=formula ADVANCE EQ SYMBOL	Defines formulas, calculates results, and inserts symbols. Equations and formulas within a document could hide sensitive information while only displaying a computed value. However, removing these may break document functionality. A special policy should specify how to handle these fields appropriately.
Index and Tables	INDEX RD TA TC TOA TOC XE	Defines entries for, and builds a Table of Contents, Table of Figures, and Table of Authorities. These fields, used for creating indexes for Tables of Contents and items associated with setting and using bookmarks, pose little threat.
Links and References	AUTOTEXT AUTOTEXTLIST HYPERLINK INCLUDEPICTURE	Inserts information from another place in the same document, from a different document or file, or from an AutoText entry. Fields, such as INCLUDETEXT and INCLUDEPICTURE, pull information from

	INCLUDETEXT LINK NOTEREF PAGEREF QUOTE REF STYLEREF	other files and display it. This applies to the fields when a site has information that changes frequently but needs to be published in several different documents. The information is maintained in one file, and other documents use the Include Text field to pull in the most recent version of the information. It is unlikely that both the documents will go through the guard. Thus, this field can pose a potential threat.
Mail Merge	ADDRESSBLOCK ASK COMPARE DATABASE FILLIN GREETINGLINE IF MERGEFIELD MERGEREC MERGESEQ NEXT NEXTIF SET SKIPIF	Defines information to use in a mail merge. Because this function includes information from a separate database into the document, leaving information about the database could pose a security issue.
Numbering	AUTONUM AUTONUMLGL AUTONUMOUT BARCODE LISTNUM PAGE REVNUM SECTION SECTIONPAGES SEQ	Specifies numbering for document items, such as sections, pages, and bar codes. These fields are keys to preserving the ability to edit the document. They present little threat.
User Information	USERADDRESS	Stores or inserts the name, initials, or address

	USERINITIALS USERNAME	of the document user.
Other	BIDIOUTLINE PRIVATE	BIDIOUTLINE sets the Right to Left for bi-directional languages such as Hebrew and Arabic. This field code may need to persist to maintain document functionality. The PRIVATE field stores data for documents that are converted from other file formats. Word will insert a PRIVATE field when converting file formats, and the field contains data needed to convert a document back to its original file format. This field could potentially hide sensitive data and would go unnoticed by manual review.

PRODUCT: WORD

LOCATION:

Field codes exist in line with the document body but the application replaces the field codes with field code-specific output. The application knows the location of all the field codes throughout the document through the use of *Plcfld* records for the Main Document, Header Document, Footnote Document, and Comment Document.

The structure of each *Plcfld* record consists of an array of character positions (CP) and *Fld* records. The *CP* records describe the index of where the field code begins within the document text. The *Fld* record contains a *flt* record, which is an index that describes the field type. Field types are any of the field codes, such as CREATEDATE, KEYWORDS, DATABASE, and NUMPAGES.

RECOMMENDATION:

WR.1 Validate: Ensure consistency and referential integrity by verifying the attributes in the *Plcfld* and *Fld* records.

WR.2 Remove: Remove the field code definition and retain the generated text by deleting the field code and adjusting the attributes in *Plcfld* and *Fld* to reflect the changes.

WR.3 Replace: Replace the generated text with administrator-defined text, and retain the field code.

WR.4 External Filtering Required: N/A

WR.5 Review: N/A

EXAMPLE:

Generally, fields aid document formatting and automation. How severely a site wants to redact fields may depend upon the importance of maintaining the document's functionality. If a document exists only for viewing, printing, and never editing again, setting all fields to *Remove* may be a valid action. However, if editing or document automation is needed in documents, one may need to set the fields to not allow removal.

OFFICE 2003.4.1: END

OFFICE 2003.4.2: Macros

DESCRIPTION:

Office includes Visual Basic support and can create everything from simple macros to data entry forms to full-blown applications. Macros allow users to automate frequently performed tasks by recording a series of commands or actions. Users may also create or modify macros manually (i.e., "by hand") with custom Visual Basic code for more advanced control. Macros can either reside locally in the application or reside in the document or template for portability.

CONCERN:

Although macros and code can be used to hide sensitive information, resulting in a data disclosure attack, the main concern with macros is the threat of malicious executable code, such as viruses, that can travel with documents. Macro viruses typically will infect only other documents and templates. Files infected with macro viruses also can affect Macintosh computers running Office software.

PRODUCT: WORD, EXCEL, POWERPOINT

LOCATION:

Macros stored in a document are represented by the Project Root Storage structure. The Project Root Storage describes a storage that contains Visual Basic for Applications (VBA) Project data. The Project Root Storage contains storage for VBA and VBA Forms, and streams for PROJECTlk, PROJECTwm, and PROJECT.

For Word documents, macros, if present, must be defined in the Project Root Storage within the file structure.

Excel documents represent the Project Root Storage by the optional *_VBA_PROJECT_CUR* record. The Project Root Storage may contain, at most, one VBA storage.

For PowerPoint documents, the Project Root Storage object is contained within either the *VbaProjectStgUncompressedAtom* or the *VbaProjectStgCompressedAtom*.

RECOMMENDATION:

AR.1 Validate: Ensure consistency and referential integrity by verifying data structures in the Project Root Storage.

AR.2 Remove: Remove from documents and templates by deleting applicable records in the Project Root Storage.

AR.3 Replace: N/A

AR.4 External Filtering Required: N/A

AR.5 Review: N/A

AR.6 White List: Retain macros in situations that require them by comparing macros to a white list of valid macros or digitally signed macros.

EXAMPLE:

For custom macros that need to survive filtering, a white list with hashes of valid macros could serve as a light-weight solution. A table of white-listed macro hash values is stored, and every macro within a given document is compared against the list. Macros that are different, even just slightly, will result in different hash values. One should remove them from the document.

OFFICE 2003.4.2: END

OFFICE 2003.4.3: Comments

DESCRIPTION:

The Comments metadata field contains author or reviewer comments. MS Office supports adding user comments to a document through the Insert → Comment command. Comments often contain private or sensitive information.

Note that another metadata item is named Comments. It is part of the Summary Properties of a document. This Comments item does not refer to the Summary Properties' Comments, but to the user comments created from the Insert → Comment command.

This item applies to MS Word, PowerPoint, and Excel (97 and higher versions).

CONCERNS:

Any free-form text fields provide potential data disclosure threats. Consider comments in Office 2003 documents a data disclosure threat.

PRODUCT: WORD

LOCATION:

The comment document contains all content in the comments. It begins at the CP, immediately following the Header Document, and is FibRgLw97.ccpAtn characters long.

A *PlcfandTxt* specifies the locations of individual comments within the comment document. The *fcPlcfandTxt* member of *FibRgFcLcb97* specifies its location. A *PlcfandRef* specifies the locations of the comment reference characters in the Main Document. The *fcPlcfandRef* member of *FibRgFcLcb97* specifies its location.

RECOMMENDATION:

WR.1 Validate: Ensure consistency and referential integrity by verifying the values in the *FibRgFcLcb97*, the *PlcfancdTxt*, and the comment document in the Header Document.

WR.2 Remove: Remove all comments via the enclosed comment document, and use the various CP records to locate and remove the references within the main document.

WR.3 Replace: N/A

WR.4 External Filtering Required: Using the CP records within the *PlcfandTxt* structure, extract the comment text, and pass the text fields to the action that is configured for handling text.

WR.5 Review: Present the text in the comments to the human reviewer.

REFERENCE:
See [MS-DOC], Section 2.3.4 Comments

See [MS-DOC], Section 2.8.7 PlcfandRef & 2.8.8 PlcfandTxt

PRODUCT: EXCEL

LOCATION:
The *NoteSh* and *NoteRR* (revision record) references the comment authors and text. The comment text is referenced within those records by its *ObjId*.

RECOMMENDATION:

ER.1 Validate: Ensure referential integrity by verifying the comment's *ObjId* presence in the applicable *NoteSh* and *NoteRR* records.

ER.2 Remove: Remove all comments by finding the *ObjId* with the appropriate "ot" enum, and remove all comments. Also remove the corresponding *NoteSh* and *NoteRR* records.

ER.3 Replace: N/A

ER.4 External Filtering Required: Using the *ObjId's* with the appropriate "ot" Note enum, scan in its "ot" attribute, scan the comment text for sensitive material, and pass the text fields to the action that is configured for handling text.

ER.5 Review: Present the comment text to the human reviewer.

REFERENCE:
See [MS-XLS], Section 2.5.185 NoteRR, & 2.5.186 NoteSh

PRODUCT: POWERPOINT

LOCATION:
The comment author, author's initials, and text are referenced in a *Comment10Containers*. Dedicated *CommentAuthorAtom*, *CommentAuthorInitialAtom*, and *CommentTextAtom* records exist within the container.

RECOMMENDATION:

PR.1 Validate: N/A

PR.2 Remove: Remove all comments by finding the *Comment10Containers* and removing the associated *Comment10TextAtom*. Also remove the *Comment10AuthorAtom* and *Comment10AuthorInitialAtom*.

PR.3 Replace: N/A

PR.4 External Filtering Required: Find all *Comment10Containers*, edit the *TabCrLfPrintableUnicodeString* in the *Comment10TextAtom*, scan the text for sensitive material, and pass the text fields to the action that is configured for handling text.

PR.5 Review: Present the text in the comments to the human reviewer.

REFERENCE:
See [MS-PPT], Sections 2.5.25 Comment10Container, 2.5.26 Comment10AuthorAtom, 2.5.27 Comment10TextAtom, 2.5.28 Comment10AuthorInitialAtom, 2.5.29 Comment10Atom.

OFFICE 2003.4.3: END

OFFICE 2003.4.4: Save History

DESCRIPTION:

This metadata includes invisible author history that contains paths or network shares. The hidden author history contains the last 10 fully-qualified path names where the document was saved. This information can provide dangerous insight into an organization's internal network.

CONCERN:

The paths and shares provide insight into the networks and paths of the organization, making them a data disclosure threat and a potential attack invitation.

PRODUCT: WORD

LOCATION:

A generic structure called *STTB* exists in Word. This structure is an array of strings. At a low - level, the save history is stored in *STTB*-style records called *SttbSavedBy*. These records store the author names, paths, and file names of the document.

See [MS-DOC].pdf, Section 2.9.292.

RECOMMENDATION:

WR.1 Validate: Ensure referential integrity by verifying that the *fcSttbSavedBy* and *lcbSttbSavedBy* of the *lcbSttbSavedBy* correctly identify the *SttbSavedBy* records.

WR.2 Remove: Remove all the save history records by deleting all of the *SttbSavedBy* records encountered. The *lcbSttbSavedBy* record contains *fcSttbSavedBy* and *lcbSttbSavedBy* attributes that contain the location and size of the *SttbSavedBy* record. Edit these attributes to reflect the deletion of the *SttbSavedBy* record. Also edit the various offset values in *FibRgFcLcb97* that follow *lcbSttbSavedBy* to account for the change.

WR.3 Replace: N/A

WR.4 External Filtering Required: Using the *fcSttbSavedBy* and *lcbSttbSavedBy* attributes of *FibRgFcLcb97*, pass any text fields in *SttbSavedBy* to the action configured for handling text.

WR.5 Review: N/A

REFERENCE:

See [MS-DOC], Section 2.2.4 STTB & 2.9.292 SttbSavedBy

OFFICE 2003.4.4: END

OFFICE 2003.4.5: Template Name

DESCRIPTION:

When using a template other than Normal.dot, the document will contain a full path to the template file. The full path can expose local path or network share information.

This issue applies to MS Word 97 and higher versions.

CONCERN:

The paths and shares provide insight into the networks and paths of the organization, making this a data disclosure threat and a potential invitation to attack.

PRODUCT: WORD

LOCATION:

The *SttbfAssoc* record is based on a *STTB* structure. It maintains the file path of the template file via the value at index 0x01.

RECOMMENDATION:

WR.1 Validate: N/A

WR.2 Remove: Remove all the template records by deleting the *SttbfAssoc* records that are found at index 0x01. The *FibRgFcLcb97* record contains *fcSttbfAssoc* and *lcbSttbfAssoc* attributes that contain the location and size of the *SttbfAssoc* record. Edit these attributes, changing them to reflect the deletion of the *SttbfAssoc* record. Also edit the various offset values in *FibRgFcLcb97* that follow *lcbSttbfAssoc* to account for the change.

WR.3 Replace: Using the *fcSttbfAssoc* and *lcbSttbfAssoc* attributes of *FibRgFcLcb97*, modify the *data* and *extraData* records of *SttbfAssoc*.

Also edit the *lcbSttbfAssoc* attribute to account for the new size of *SttbfAssoc,* and the various offset values in *FibRgFcLcb97* that follow it, to account for the change. Also edit the *lcbSttbfAssoc* attribute to account for the new size of *SttbfAssoc,* and the various offset values in *FibRgFcLcb97* that follow it, to account for the change.

WR.4 External Filtering Required: Using the *fcSttbfAssoc* and *lcbSttbfAssoc* attributes of *FibRgFcLcb97*, pass text fields in *SttbfAssoc* to the action that is configured for handling text.

WR.5 Review: Present the template to the human reviewer.

REFERENCE:

See [MS-DOC], Sections 2.9.273 SttbfAssoc

OFFICE 2003.4.5: END

OFFICE 2003.4.6: Scenarios

DESCRIPTION:

Scenarios are an Excel feature that allow multiple data models. MS Excel supports entering

multiple data models within specific areas of a spreadsheet (Tools → Scenario). After selecting a scenario, the remaining scenarios may expose data models that should not be exposed after document release to an outside party. This issue applies to MS Excel 97 and higher versions.

CONCERN:

Any free-form text field provides potential data disclosure threats. Consider data fields (like those in scenarios) in Excel documents a data disclosure threat.

PRODUCT: EXCEL

LOCATION:

The data structure *ScenMan* represents a group of scenarios, where the *isctCur* attribute represents the currently selected *Scenario*. The attribute *isctShown* contains the index of the *Scenario* that calculated the current results. Because of this index, data in the selected *Scenario* may be out of sync with the applied scenario. If this situation is not accounted for, it may provide another means for delivering more data than intended.

RECOMMENDATION:

ER.1 Validate: Ensure referential integrity by tracing the *ScenMan* attributes and verifying that all indices point to valid scenarios. Also ensure referential integrity by scanning for scenarios that are not indexed in the *ScenMan* record.

ER.2 Remove: Remove all scenarios by deleting all data in the *Scenario* and *ScenMan* records. Remove all non-selected scenarios by referencing the *ScenMan* records. The data in the actual cells remain untouched.

ER.3 Replace: Resolve potential current-scenario/shown-scenario conflicts in favor of the current scenario by adjusting all *isctShown* attributes to match *isctCur*.

ER.4 External Filtering Required: Find each *Scenario* via the *ScanMan* record. Scan the text of the *rgchName, rgchNameUser*, and *rgchComment* attributes and pass the data to the action that is configured for text data.

ER.5 Review: Find each Scenario via the *ScanMan* record. Scan the text of the *rgchName*, *rgchNameUser*, and *rgchComment* attributes and present the content to the human reviewer.

REFERENCE:

See [MS-XLS], Sections 2.4.244 SCENARIO, & 2.4.246 ScenMan

OFFICE 2003.4.6: END

OFFICE 2003.4.7: Hyperlinks

DESCRIPTION:

This item includes hyperlinks that contain either fully-qualified local paths, or network share names, or e-mail addresses. The Office hyperlink feature (Insert → Hyperlink) allows the creation of links to various locations.

CONCERN:

Hyperlinks can pose an attack risk and a data disclosure risk. The attack risk can occur through

hyperlinks that take the user to malicious sites or attempt to launch dangerous code on the local machine.

Data disclosure can occur through the several possibilities: fully qualified local paths, network paths, and e-mail addresses. These paths can provide unwanted insight into an organization's internal structure.

Although this document focuses on hyperlink data structures within each file format, it is important to note that plain text hyperlinks and e-mail addresses may require performing the same filtering actions.

PRODUCT: WORD

LOCATION:

Hyperlinks in Word primarily exist as Field Codes within the *WordDocumentStream*. A sample inline hyperlink located in a Word document follows the structure below:

HYPERLINK "http://www.google.com" www.google.com

For more details on Field Codes, see OFFICE 2003.3.1 Field Codes.

PRODUCT: EXCEL

LOCATION:

Hyperlinks in Excel exist as *HLink* records. The important data in a *HLink* record are the *ref8* structure, which identifies the range of cells containing the hyperlink, and the hyperlink object, which specifies the actual hyperlink. This hyperlink object has multiple locations in which URLs, pathnames or server names may exist. Most notably are the *displayName*, *moniker*, *location* and *oleMoniker* variables within the hyperlink object, which all may hold link data. *displayName*, *moniker*, and *location* are of type *HyperlinkString*. Structural details of *HyperlinkString* can be found in [MS-OSHARED], Section 2.3.7.9. *oleMoniker* is of type *HyperlinkMoniker*. More information of this structure can be found in [MS-OSHARED], Section 2.3.7.2.

For more details on the *hyperlink* object, see [MS-OSHARED], Section 2.3.7.1.

PRODUCT: POWERPOINT

LOCATION:

Hyperlinks in PowerPoint use the *ExHyperlinkContainer* record. The *ExHyperlinkContainer* (See [MS-PPT], Section 2.10.16) holds a collection of atoms including *exHyperlinkAtom*, *friendlyNameAtom*, *targetAtom* and *locationAtom*. The latter three atoms each contain a single Unicode string which describes some detail of the hyperlink. For example, *friendlyNameAtom* (See [MS-PPT], Section 2.10.18) specifies a user-readable name of the hyperlink; *targetAtom* (See [MS-PPT], Section 2.10.19) specifies the full path of the hyperlink destination file; and *locationAtom* (See [MS-PPT], Section 2.10.20) specifies the location of the hyperlink within the destination file.

PRODUCT: WORD, EXCEL, POWERPOINT

LOCATION:

Additionally, hyperlinks in Word, Excel, and PowerPoint documents may reside in *VtHyperlinks* property structures to which the *_PID_HLINKS* variable refers. The *VtHyperlinks* property is a collection of all the hyperlinks within a document. Although all three formats may use this property structure, each file format primarily relies on its aforementioned data structure for handling hyperlinks. The discussion of the *VtHyperlinks* property below gives sanitization and

filter developers a more thorough description of possible hyperlink locations.

A *VtHyperlinks* property has an embedded *VtHyperlinkValue* structure, which is referred to as *vtValue*. This variable contains an embedded *VecVtHyperlink* structure, which is referred to as *vecHyperlink*. The *VecVtHyperlink* structure contains an array of *VtHyperlink* structures for each hyperlink in the document, which is referred to by the variable name *rgHyperlink*.

The array of *VtHyperlink* structures contains all the hyperlinks in the document, which are referred to sequentially as *linkElement-1*, *linkElement-2*… *linkElement-n*. Each *VtHyperlink* contains six structures that define where the hyperlink applies in the document and to where each hyperlink links. Location values reside in variables *hlink1* and *hlink2*. Each *VtHyperlink* also contains a calculated hash of each hyperlink's two location values, *hlink1* and *hlink2*, in a property called *dwHash*.

The values of *hlink1* and *hlink2* differ based on the type of hyperlink. The value of *hlink1* refers to an external target, such as a URL or file system path. The value of *hlink2* refers to a specific item within *hlink1*, such as a target or bookmark.

The *dwHash* property is calculated according to the formula outlined in Section 2.4.2, Hyperlink Hash of [MS-OSHARED].pdf.

RECOMMENDATION:

AR.1 Validate: N/A

AR.2 "Remove: Remove only hyperlinks designated as "sensitive". The minimal sensitive hyperlink setting should clean (remove) only hyperlinks that meet the following criteria:

Fully qualified local path (, i.e., it must start with "//")

Network share name (This only works in PowerPoint, which stores more information about the hyperlinks than Excel or Word.)

A string, which matches a regular expression. For example:

*.*yahoo.** <*space*> *.*msn.** <*space*> *.*google.** <*space*> *^https://.** "

AR.3 Replace: Replace hyperlinks that meet one or many of the following criteria, with a predefined value:

Fully qualified local path (, i.e., it must start with "//")

Network share name (This only works in PowerPoint, which stores more information about the hyperlinks than Excel or Word.)

A string, which matches a regular expression. For example:

*.*yahoo.** <*space*> *.*msn.** <*space*> *.*google.** <*space*> *^https://.** "

AR.4 External Filtering Required: Pass hyperlink text to the action that is configured for text.

AR.5 Review: Present to the human reviewer the list of hyperlinks and all associated text.

REFERENCE:
See [MS-OSHARED], Office Common Data Types and Objects Structure Specification.

OFFICE 2003.4.7: END

OFFICE 2003.4.8: Summary Properties

DESCRIPTION:

Word, Excel, and PowerPoint formats support the Document properties feature to allow additional information about the document (metadata) to reside in the file container. A predefined list of properties exists for each document format and all three document formats share some common properties, such as *Author* and *Last Save Time*.

To view Summary Properties in the document, open the document in its application (Word, PowerPoint, or Excel) and select File → Properties → Summary.

Table 4-2. Summary Properties

Summary Properties	Applicable to:		
	Word	PPT	Excel
Title	X	X	X
Subject	X	X	X
Author	X	X	X

Manager	X	X	X
Company	X	X	X
Category	X	X	X
Keywords	X	X	X
Comments	X	X	X
Template	X	X	X
Thumbnail	X	X	X
Last Author	X	X	X
Revision Number	X	X	X
Total Editing Time	X	X	X
Last Printed Time	X	X	X
Create Time	X	X	X
Last Saved Time	X	X	X
Num Pages	X	X	X
Num Words	X	X	X
Num Chars	X	X	X
Thumbnail info	X	X	X
Name Of Creating Application	X	X	X
Security	X	X	X

CONCERN:
Summary properties are pre-defined, application-specific properties. They pose a data disclosure risk because they remain obscured from typical reading and writing activities of document editing. One can overlook these properties and sensitive information potentially could escape through these fields.

PRODUCT: WORD, EXCEL, POWERPOINT

LOCATION:
Document properties, both Summary Properties and Custom Properties, exist in structures called *PropertySetStreams*. The *PropertySetStream* contains an array of *PropertySet* structures. Typically, each *PropertySetStream* only contains one *PropertySet*; however, in special cases, such as with the *DocumentSummaryInfo* and *UserDefinedProperties* property sets, a *PropertySetStream* will contain multiple *PropertySets*. Summary Property, *PropertySetStreams*, exists in the document stream named "*\005SummaryInformation*". The exception to this is the *Title Summary* Property, which also exists in the document stream "*\005DocumentSummaryInformation*."

The *PropertySet* structure contains an array of *PropertyIdentifierAndOffset* and an array of *Property* objects. Each *PropertyIdentifierAndOffset* object contains a *PropertyIdentifier* and *Offset* value. The *PropertyIdentifier* value can identify a reserved generic property name, such as *PIDSI_AUTHOR*, *PIDSI_KEYWORDS*, or and *PIDSI_COMMENTS*.

The *PropertySet's* Property object contains either a *TypedPropertyValue* object or a *Dictionary* object. The *Dictionary* object is used for custom properties that users create.

The *TypedPropertyValue* object contains the actual values of the generic properties. This object contains three values: (1) the *Type*, which refers to the format of the property; (2) the *Padding*, two

bytes of zeros; and (3) the *Value*, the user's input value that is represented and serialized according to the value of the *Type* variable.

RECOMMENDATION:

AR.1 Validate: Ensure consistency and referential integrity by verifying the data structures within the *PropertySteStream* and verifying the relationships between the parent and child data structures used to represent document properties.

AR.2 Remove: Remove the entire contents of the field by writing 0x00 or 0xFF to the fields.

AR.3 Replace: Replace the value with replacement text. Ensure that the whole field is overwritten.

AR.4 External Filtering Required: Pass all text fields to the action that is configured for handling text.

AR.5 Review: N/A

OFFICE 2003.4.8: END

OFFICE 2003.4.9: Custom Properties

DESCRIPTION:

Word, Excel, and PowerPoint formats support the Document properties feature to allow additional information about the document (metadata) to reside in the file container. In addition to the built-in Summary Properties, Custom Properties provide additional storage of named value pairs. Each document can have its own unique set of custom properties, as defined by the document editor.

To view Custom Properties in the document, open the document in its application (Word, PowerPoint, or Excel) and select File → Properties → Custom.

CONCERN:

Custom Properties are user-defined properties that do not follow a pre-set pattern. These properties pose a data disclosure risk because the Document Properties, including the Custom Properties, are obscured from the typical reading and writing activities of document editing. Often, one can overlook these properties and sensitive information could potentially escape through these fields.

PRODUCT: WORD, EXCEL AND POWERPOINT

LOCATION:

Document properties, both Summary Properties and Custom Properties, exist in structures called *PropertySetStream*s. The *PropertySetStream* contains an array of *PropertySet* structures. Typically,

each *PropertySetStream* only contains one *PropertySet***.** However, in special cases, such as with the *DocumentSummaryInfo* and *UserDefinedProperties* property sets, a *PropertySetStream* will contain multiple *PropertySet*s. Custom Property, *PropertySetStream*s, exist in the document stream named "*\005DocumentSummaryInformation*"**.**

The *PropertySet* structure contains an array of *PropertyIdentifierAndOffset* and an array of *Property* objects. Each *PropertyIdentifierAndOffset* object contains a *PropertyIdentifier* and *Offset* value. The *PropertyIdentifier* value can identify a reserved generic property name, such as *PIDSI_AUTHOR*, *PIDSI_KEYWORDS* and *PIDSI_COMMENTS*, or it can identify custom properties through the use of the dictionary identifier, known as *DICTIONARY_PROPERTY_IDENTIFIER*.

The *PropertySet*'s Property object contains either a *TypedPropertyValue* object or a *Dictionary* object. If the *DICTIONARY_PROPERTY_IDENTIFIER* was referenced in the *PropertyIdentifierAndOffset* array previously, the *Dictionary* object stores the names of the custom properties that the user creates.

The *Dictionary* object contains an array of *DictionaryEntry* objects, each with their own *PropertyIdentifier*, *Length***,** and *Name* variables. Each *DictionaryEntry* object only stores the name of the custom property. The corresponding custom value is stored in a subsequent *TypedPropertyValue* object.

The *TypedPropertyValue* object contains the actual values of the generic and custom properties. This object contains three values: (1) the *Type*, which refers to the format of the property; (2) the *Padding*, two bytes of zeros; and (3) the *Value*, the user's input value that is represented and serialized according to the value of the *Type* variable.

RECOMMENDATION:

AR.1 Validate: Ensure consistency and referential integrity by verifying the data structures within the *PropertySteStream* and verifying the relationships between the parent and child data structures that are used to represent document properties.

AR.2 Remove: Remove the entire field contents by writing 0x00 or 0xFF to the field.

AR.3 Replace: Replace the value with replacement text. Ensure that the whole field is overwritten**.**

AR.4 External Filtering Required: Pass all text fields to the action that is configured for handling text.

AR.5 Review: N/A

AR.6 White List: Provide a white list of acceptable Custom Property names. Checking the Custom Properties of the target document will allow specific custom properties to pass through the inspection or sanitization process intact.

EXAMPLE:
An example Custom Property white list is as follows:

Table 4-3. Custom Property White List Example

Sample Strings	Explanation
Project: Red Project: Blue	Either Red or Blue will be accepted for the Project, but no other values will be accepted.

Destination	The Destination Property will be accepted with any value.
Checked by	The Checked by Property will be accepted with any value.

OFFICE 2003.4.9: END

OFFICE 2003.4.10: Footnotes and Endnotes

DESCRIPTION:

Footnotes and endnotes are structures in Word for displaying references at the end of the document or at the bottom of a page. Selecting Insert → Reference → Footnote creates them.

CONCERN:

Any free-form text field provides the possibility of data disclosure threats. Consider footnotes and endnotes in Office 2003 documents a data disclosure threat.

LOCATION:

The footnote document contains all content for footnotes and endnotes. It begins immediately after the main document and is *FibRgLw97.ccpftn* bytes long.

RECOMMENDATION:

WR.1 Validate: Ensure referential integrity by verifying the presence of the footnote document following the main document and checking that it is *FibRgLw97.ccpftn* bytes long.

WR.2 Remove: Remove all footnotes and endnotes via the enclosed footnote document, and use the various CP records to locate and remove the references within the main document.

WR.3 Replace: In the footnote document, modify the text of each footnote and endnote to edit the material.

WR.4 External Filtering Required: Pass the text in the footnotes document to the action that is configured for text.

WR.5 Review: N/A

OFFICE 2003.4.10: END

OFFICE 2003.4.11: Headers and Footers

DESCRIPTION:

These options check the target document's headers and footers for classification markings. When used, documents will fail if they do not have headers/footers that meet the defined requirements. This includes failing documents without headers/footers. The requirement is that each header/footer in the document must contain at least one of the expressions in the header/footer *Must Include* list.

CONCERN:

Any free-form text field provides potential data disclosure threats. Consider headers and footers in Office 2003 documents a data disclosure threat.

PRODUCT: WORD

LOCATION:

The header document contains all content in headers, footers, and footnote and endnote separators. It begins immediately after the footnote document and is *FibRgLw97.ccpHdd* bytes long.

The header document is split into text ranges called *stories*, as specified by *PlcfHdd*. Each story specifies the contents of a single header, footer, or footnote/endnote separator. If a story is non-empty, it MUST end with a paragraph mark that serves as a guard between stories. Numerous reserved stories exist within the headers and footers for formatting, all separated by paragraph marks.

However, it may prove beneficial to consider all individual stories as potential targets, rather than trying to eliminate all the formatting stories based simply on their position within the range.

Note: Contrary to Microsoft's documentation on the subject, even and first page headers/footers are not necessarily empty when not in use. For example, turn on even/odd page headers and modify the even page header. Then turn off even/odd page headers. The even page headers have been replaced with the same headers as the odd pages. When one saves the document, the even page header information remains.

RECOMMENDATION:

WR.1 Validate: Ensure referential integrity by verifying the presence of the header document after the footnote document and checking that it is *FibRgLw97.ccpHdd* bytes long.

WR.2 Remove: Remove all headers and footers via the enclosed header documents, and use the various CP records to locate and remove the references within the main document.

WR.3 Replace: Using the CP records within the *Plcfhdd* and *PlcfndTxt* structures, modify the text of the headers and footers to edit the material. Remove found source material.

WR.4 External Filtering Required: Present the text to the action that is configured for text.

WR.5 Review: N/A

REFERENCE:

See [MS-DOC], Sections 2.3.4 Comments, 2.8.7 PlcfandRef, & 2.8.8 PlcfandTxt

PRODUCT: EXCEL

LOCATION:
There are two header/footer records of interest in Excel.

Header: This record specifies the header text of the current sheet, when printed, and contains a block of *XLUnicodeString* that contains the text. Various reserved tags for formatting are embedded in the string.

HeaderFooter: This record contains various blocks of interest, each containing *XLUnicodeStrings*, including *strHeaderEven*, *strFooterEven*, *strHeaderFirst*, and *strFooterFirst*.

RECOMMENDATION:

ER.1 Validate: N/A

ER.2 Remove: Remove all headers and footers by replacing the various "*str*" attributes with empty *XLunicodeString* values. Also, change the various corresponding "*cch*" attributes to account for an adjusted size of 0.

Find all headers and footers within the *HeaderFooter* record. Edit the text of each "*str*" attribute. Also, change the various corresponding "*cch*" attributes to account for an adjusted size of the new content. Also, change the various corresponding "*cch*" attributes to account for an adjusted size of 0.

ER.3 Replace: N/A

ER.4 External Filtering Required: Present the text to the action that is configured for text.

ER.5 Review: N/A

REFERENCE:
See [MS-XLS], Sections 2.4.136 Header, 2.4.137 HeaderFooter, 2.5.134 FrtFlags, & 2.5.135 FrtHeader

PRODUCT: POWERPOINT

LOCATION:
The text of a PowerPoint header and footer is represented as a block of *PrintableUnicodeString*, and resides in the header and footer sections of the *HeaderAtom* and *FooterAtom,* respectively. These atoms are referenced in *NotesHeadersFootersContainer*, *PerSlideHeadersFootersContainer***,** and *SlideHeadersFootersContainer*.

RECOMMENDATION:
PR.1 Validate: Ensure referential integrity by verifying the references to *HeaderAtom* and *FooterAtom* in *NotesHeadersFootersContainer*, *PerSlideHeadersFootersContainer***,** and *SlideHeadersFootersContainer*.

PR.2 Remove: Remove all headers and footers by finding the *SlideHeadersFootersContainers*, *NotesHeadersFootersContainers*, and *PerSlideHeadersFootersContainers*. Remove them and all contained records.

PR.3 Replace: N/A

PR.4 External Filtering Required: Present the text to action that is configured for text.

REFERENCE:
See [MS-PPT], Section 2.4.15 Bar

OFFICE 2003.4.11: END

OFFICE 2003.4.12: Hidden Text

DESCRIPTION:

Note: For the purposes of this section, "Hidden Data" does not refer to the many ways to obfuscate or bury data in Office 2003 files. It is specific to the hidden text/cell/slide functions readily available in the Office 2003 Graphical User Interface (GUI).

In Word 2003, this metadata item includes text that the author hid.

Text that the user intentionally formatted as hidden (Format → Font → Font tab → Hidden effects checkbox) may contain sensitive information that one should review or remove before distributing the document.

In Excel 2003, this item refers to hidden Excel spreadsheet rows, columns, or worksheets. Hidden cells may contain sensitive data that requires user review prior to release. Hidden cells can be identified during analysis, but require user review before being deleted or cleared, because they may be required to resolve references from visible cells.

In PowerPoint 2003, this item included slides that were hidden from presentation and printing. The PowerPoint hidden slide feature (Slide Show → Hide Slide) allows hiding individual slides during the slide show and presentation printing. Hidden slides may contain information that is not intended for general release.

CONCERN:

Any free-form text field provides potential data disclosure threats. Consider hidden data in Office 2003 documents a data disclosure threat.

PRODUCT: WORD

LOCATION:

Hidden in Word is just another style, akin to italic and bold. Like having a row of characters "i**n** b**old** a**n**d **p**l**ain** te**x**t" with alternating styles, hidden characters can exist among visible ones. It is considered an attribute of each individual character in the set.

Character attributes in Word 2003 are referred to as Single Property Modifiers (sprm). The

attributes of interest for hidden data are *sprmCFFIdVanish*, *sprmCFVanish*, and *sprmCFWebHidden* (to a lesser extent). Note that the means to hide text using other character attributes, such as matching their colors with the background color, also exists and is covered under different topics.

Another consideration point is the application behavior of *sprmCFVanish*, in that it often unhides or undoes the effects of a previous *sprmCFVanish*. This makes the behavior of the attribute more contextual. Is the *sprmCFVanish* hiding content, or is it instead exposing a portion of content *within* a block of hidden content?

RECOMMENDATION:

WR.1 Validate: N/A

WR.2 Remove: Remove text with hiding attributes (*sprmCFFIdVanish, sprmCFVanish*, and *sprmCFWebHidden*), set by scanning all Single Property Modifiers, looking for uses of these attributes. Note that that the purpose of the *Vanish* attribute may be to expose (not hide) content.

WR.3 Replace: Replace the hidden text or modify the attribute to expose the hidden text by scanning all Single Property Modifiers, and looking for uses of the attributes (*sprmCFFIdVanish, sprmCFVanish*, and *sprmCFWebHidden*). Note that the purpose of the Vanish attribute may be to expose (not hide) content.

WR.4 External Filtering Required: Present the hidden text to the action that is configured for text.

WR.5 Review: Extract and present hidden text for review by scanning all Single Property Modifiers, and looking for uses of hiding attributes (*sprmCFFIdVanish, sprmCFVanish*, and *sprmCFWebHidden*). Note that the purpose of the Vanish attribute may be to expose (not hide) content.

REFERENCE:
See [MS-DOC], Sections 1.3.3 Formatting & 2.6.1 Character Properties

PRODUCT: EXCEL

LOCATION:
As Word documents consider the style of each individual character in a document, Excel also maintains style attributes for individual cells. Excel, however, maintains attributes at various scope levels (like attributes for an entire sheet) to consider also.

BoundSheet8 records contain information about a sheet. It includes the *hsState* attribute. This attribute controls the hidden status of an entire sheet, including if the sheet is hidden so that the Excel GUI cannot make it visible.

ColInfo records contain formatting for a set of columns, and include the *ixfe* structure. The *ixfe* structure's first bit is the *fHidden* attribute, indicating if the column's default state is or is not hidden.

Lbl records contain information on defined names, or tags, to clarify the purpose of an item in Excel. *Lbl* records include an *fHidden* bit that controls if the defined name is visible in a list along with other defined names. A revision record could also exist for a defined name. *RRDDefNameFlags*, an *fHidden* attribute, also exists in the record.

The *Row* record is the equivalent of *ColInfo* for rows. It contains a *fDyZero* attribute that controls if a row is hidden.

Excel *Scenario* records contain an *fHidden* attribute that controls if Excel scenarios remain hidden in

a protected workbook.

StyleExt (an extension of *Style*) records contain additional style information for individual cells and hold an *fHidden* attribute. An *XFProp* structure exists, which contains an *xfPropDataBlob* variable-length field that could indicate hidden status also. See *XF*.

SXAddl_SXCField_SXDVer10Info records contain additional configuration data regarding PivotTables. The *fHideDD* attribute controls if the per pivot drop-down GUI remains hidden for the pivot field.

SXAddl_SXCField12_SXDVer12Info records contain additional configuration data regarding PivotTables. The *fHiddenLvl* attribute controls if the Online Analytical Processing (OLAP) pivot field remains a hidden level.

SXAddl_SXCHierarchy_SXDInfo12 records contain additional configuration data regarding PivotTables. The *fHidden* attribute controls if the OLAP hierarchy corresponding to the pivot hierarchy remains hidden.

SXAddl_SXCView_SXDVer10Info records contain additional configuration data regarding PivotTables. The *fHideDDData* attribute controls if control for selecting the pivot items to display in the PivotTable view remains hidden.

SXAddl_SXCView_SXDVer12Info records contain additional configuration data regarding PivotTables. The *fHideDrillIndicators* attribute controls if the control for expanding or collapsing inside of a pivot item remains hidden.

cHiddenMemberSets and *rgHiddenMemberSets* are attributes of *SXTH***.** They control which Pivot Hierarchies in the PivotTable remain hidden.

SXVDEx is an extension record for Pivot field properties. It contains an *fHideNewItems* attribute, and controls if new pivot items appear, after a refresh, hidden by default.

SXVI contains Pivot item properties and contains an *fHidden* attribute. It controls if the Pivot Item remains visible.

UserBView is a record that specifies the custom view settings for the workbook. It includes a *fRowColIncl* attribute, indicating if the view includes hidden rows, columns and filters.

UserSViewBegin is a record that specifies custom view settings for a specific sheet in the workbook. It contains a *fFilterMode* attribute, indicating if the view has hidden cells due to filtering. The *fHiddenRw* and *fHiddenCol* attributes indicate the state of hidden rows and columns, respectively.

UserSViewBegin_Chart performs the same function for chart sheets as *UserSViewBegin* (above). It contains an *hsState* attribute that controls its visibility.

Window1 contains attributes of a window to display a sheet. It contains *fHidden*, which works similar to the other *fHidden* attributes listed here, but also includes *fVeryHidden*. Other uses of *fHidden* include a setting that indicates (in effect) the user remains unaware of a hidden item. The *fVeryHidden* attribute has that functionality separated so that bits can represent both attributes.

When using graphical icons to represent data in a particular range, the underlying data still exists but may remain hidden from view. The *CFMultistate* record has a *fIconOnly* attribute that controls that behavior.

DXFProt has an *fHidden* attribute, but the meaning differs somewhat from other instances. Similar to the way *fHidden* functions in Scenario, it controls if cells remain visible in a protected workbook.

NoteRR is a comment revision record. It contains *fRwHidden* and *fColHidden* attributes that control if rows or columns of the Note revision record remain hidden.

NoteSh has the same attributes for the Note.

RECOMMENDATION:

ER.1 Validate: N/A

ER.2 Remove: Remove hidden text by scanning all the listed hiding attributes.

ER.3 Remove: Remove hidden sheets by examining the appropriate records and attributes to locate hidden sheets.

ER.4 Remove: Remove hidden columns and rows by examining the appropriate records and attributes to locate hidden columns or rows.

ER.5 Remove: Remove hidden cells by examining the appropriate and records to locate hidden cells.

ER.6 Replace: Replace all hidden text by scanning all the listed hiding attributes and changing the attributes to expose the content.

ER.7 Replace: Replace hidden sheets by scanning for the appropriate records and attributes and changing the values to expose the sheet.

ER.8 Replace: Replace hidden columns and rows by scanning for the appropriate records and attributes and changing the values to expose the columns and rows.

ER.9 Replace: Replace hidden cells by scanning for the appropriate records and attributes and changing the values to expose the cells.

ER.10 External Filtering Required: Present the hidden text to the action that is configured for text.

ER.11 Review: Scan for hidden data (sheets, rows, columns, cells and text) and present it for review.

REFERENCE:

[MS-XLS] PowerPoint Binary File Format (.ppt) Structure Specification

PRODUCT: POWERPOINT

LOCATION:

SlideShowSlideInfoAtom is a record that (by name) contains information on slide transitions. However, this record also contains an *fHidden* attribute that controls if the slide immediately following a transition displays during a slide show.

Animation attributes resides in the *AnimationInfoAtom* record. One of the attributes, **fHide**, hides the shape/media when it is not playing.

PrintOptionsAtom contains an attribute, named *fPrintHidden***.** It controls if hidden slides are printed.

RECOMMENDATION:

PR.1 Validate: N/A

PR.2 Remove: Remove all hidden slides by scanning all instances of *SlideShowSlideInfoAtom*, looking for uses of the *fHidden* attribute. If the attribute is used, delete the following slide and

adjust the *fHidden* attribute to compensate.

PR.3 Replace: Replace the hidden attributes (expose the slide) by scanning all instances of *SlideShowSlideInfoAtom*, looking for uses of the *fHidden* attribute.

PR.4 External Filtering Required: Present the text of hidden slides to the action that is configured for text.

PR.5 Review: Pass all hidden slides for review.

REFERENCE:
See [MS-PPT], Section 2.6.6 SlideShowSlideInfoAtom, Section 2.8.2 AnimationInfoAtom, Section 2.4.12 PrintOptionsAtom

OFFICE 2003.4.12: END

OFFICE 2003.4.13: Authors

DESCRIPTION:
Hidden author history is contained in this metadata field within MS Word documents. Up to the last 10 authors that saved the document reside in an area of the document that one cannot access using the Word application. In Word 97 and Word 2000, this information also contains the save paths for the document and may include sensitive user logon or network share information.

Notes: Consider that Comments, in particular, carry author information also. See Comments for details. Also, consider that author names reside in the Save history. See Author History Save Paths for details.

CONCERN:
Any free-form text fields provide potential data disclosure threats. Consider headers and footers in Office 2003 documents s a data disclosure threat.

PRODUCT: WORD

LOCATION:
In Word, a generic structure called *STTB* exists. This structure is an array of strings. At a low - level, the author's information resides in *STTB*-style records called *SttbfRMark*. These records store the authors for revisions, emails, and comments. Numerous places exist in Word documents to hold indexes to particular positions in the *SttbfRMark* record.

Word also contains structures that are specific to containing email information. The *RmdThreading* record contains *SttbAuthorAttrib* and *SttbAuthorValue* records to consider.

The *SttbfAssoc* record is based on a *STTB* structure. It maintains the document author at index *0x06* and the last user who changed the document at *0x07*.

The documents summary information is a property set containing *PIDSI_AUTHOR*. The attribute contains a short author name in the *Characters* sequence.

The documents summary information is a property set containing *PIDSI_LASTAUTHOR*. The attribute contains an author name in the *Characters* sequence.

RECOMMENDATION:

WR.1 Validate: Ensure referential integrity by verifying the indices into *SttbfRMark* and *SttbAuthor* and *Attrib SttbAuthorValue* in *RmdThreading*.

WR.2 Remove: Remove all author records by doing the following:

Deleting all encountered *SttbfMark* records. The *FibRgFcLcb97* record contains *fcSttbfMark* and *lcbSttbfMark* attributes that contain the location and size of the *SttbfMark* record. Edit these attributes to reflect the deletion of the *SttbfMark* records. Also, edit the various offset values in *FibRgFcLcb97* that follow *lcbSttbfMark* to account for the change.

Using the *fcSttbfMark* and *lcbSttbfMark* attributes of *FibRgFcLcb97*, modify the *data* and *extraData* records of *SttbfMark* to edit the material. Also, edit the *lcbSttbfMark* attribute to account for the new size of *SttbfMark,* and the various offset values in *FibRgFcLcb97* that follow it, to account for the change.

Remove all author records by deleting all encountered *RmdThreading* records. The *FibRgFcLcb2000* record contains *fcRmdThreading* and *lcbRmdThreading* attributes, which contain the location and size of the *RmdThreading* record. Edit these attributes to reflect the deletion of the *RmdThreading* records. Also, edit the various offset values in *FibRgFcLcb2000* that follow *lcbRmdThreading* to account for the change.

Remove the *PIDSI_AUTHOR* and *PIDSI_LASTAUTHOR* records of the *SummaryInformation* property set.

WR.3 Replace: Replace the author by doing the following:

Deleting all encountered *SttbfMark* records. The *FibRgFcLcb97* record contains *fcSttbfMark* and *lcbSttbfMark* attributes that contain the location and size of the *SttbfMark* record. Edit these attributes to reflect the deletion of the *SttbfMark* records. Also, edit the various offset values in *FibRgFcLcb97* that follow *lcbSttbfMark*, to account for the change.

Using the *fcSttbfMark* and *lcbSttbfMark* attributes of FibRgFcLcb97, modify the *data* and *extraData* records of *SttbfMark* to edit the material. Also, edit the *lcbSttbfMark* attribute to account for the new size of *SttbfMark,* and the various offset values in *FibRgFcLcb97* that follow it, to account for the change.

Remove all author records by deleting all encountered *RmdThreading* records. The *FibRgFcLcb2000* record contains *fcRmdThreading* and *lcbRmdThreading* attributes, which contain the location and size of the *RmdThreading* record. Edit these attributes to reflect the deletion of the *RmdThreading* records. Also, edit the various offset values in *FibRgFcLcb2000,* that follow *lcbRmdThreading*, to account for the change.

Remove the *PIDSI_AUTHOR* and *PIDSI_LASTAUTHOR* records of the *SummaryInformation* property set.

WR.4 External Filtering Required: N/A

WR.5 Review: N/A

REFERENCE:
See [MS-DOC], Sections 2.2.4 STTB, 2.9.227 RmdThreading, 2.9.273 SttbfAssoc, & 2.9.287

SttbfRMark

See [MS-OLEPS], Sections 3.1.4 PIDSI_AUTHOR & 3.1.8 PIDSI_LASTAUTHOR

PRODUCT: EXCEL

LOCATION:
In shared workbooks, where multiple parties can modify a single workbook, another set of records exists for tracking use. *UsrInfo* records contain information about which users have a workbook open at a given point and time.

In *Scenario* records, an *rgchNameUser* attribute exists to store the author of the given scenario.

RECOMMENDATION:

ER.1 Validate: N/A

ER.2 Remove: Remove author information by deleting or nullifying *UsrInfo* records and the *rgchNameUser* field of *Scenario* records.

ER.3 Replace: Replace author information by modifying out *UsrInfo* records and the *rgchNameUser* field of *Scenario* records.

ER.4 External Filtering Required: N/A

ER.5 Review: N/A

REFERENCE:
See [MS-XLS], Sections 2.4.244 SCENARIO, 2.4.340 UsrInfo, 2.5.239 ShortDTR, & 2.5.241 SLCO8

PRODUCT: POWEROINT

LOCATION:

The *CurrentUserAtom* record contains information regarding the last user to modify the presentation, including the *ansiUserName* and *unicodeUserName* attributes.

The *BCUserNameAtom* record contains an attribute called *username*. The record stores the name of a user who scheduled a presentation broadcast.

RECOMMENDATION:

PR.1 Validate: N/A

PR.2 Remove: Remove author information by nullifying the *ansiUserName* and *unicodeUserName* fields in *CurrentUserAtom* and the *username* attribute in *BCUserNameAtom*.

PR.3 Replace: Replace author information by modifying the *ansiUserName* and *unicodeUserName* fields in *CurrentUserAtom* and the *username* attribute in *BCUserNameAtom*.

PR.4 External Filtering Required: N/A

PR.5 Review: N/A

REFERENCE:
See [MS-PPT], Sections 2.3.2 CurrentUserAtom & 2.4.17.17 BCUserNameAtom

OFFICE 2003.4.13: END

OFFICE 2003.4.14: Routing Slip

DESCRIPTION:

This item includes e-mail routing information. The e-mail routing feature of MS Office (File → Send To → Routing Recipient) stores the email addresses and user names of recipients in the document.

This item applies to MS Word, PowerPoint, and Excel (97 and higher versions).

CONCERN:

E-mail addresses and names carry a data disclosure threat.

PRODUCT: WORD

LOCATION:

The *RouteSlipInfo* structure contains an attribute called *szName*, which contains the recipient's name or e-mail address.

RECOMMENDATION:

WR.1 Validate: N/A

WR.2 Remove: Remove all routing slip records by deleting all of the *RouteSlipInfo* records in the *RouteSlip* structure. Adjust the *fcRouteSlip* and *lcbRouteSlip* fields and other effected offsets in *FibRgFcLcb97* accordingly.

WR.2 Replace: Replace routing slip information by isolating all of the *RouteSlipInfo* records in the *RouteSlip* structure, replace the *szName* attributes with the specified text, and alter the *cbszName* attribute with the field's new size. Adjust the *fcRouteSlip* and *lcbRouteSlip* fields and other affected offsets in *FibRgFcLcb97* accordingly.

WR.4 External Filtering Required: Present any text in the routing slip records to the action that is configured for text.

WR.5 Review: N/A

REFERENCE:

See [MS-DOC], Section 2.9.230 RouteSlipInfo

PRODUCT: EXCEL

LOCATION:

The *DocRoute* record contains an attribute called *rgchSSAddr*. This attribute identifies the originator's e-mail address. The *RecipName* record also contains *rgchSSAddr* for the recipient's address.

RECOMMENDATION:

ER.1 Validate: N/A

ER.2 Remove: Remove e-mail addresses by isolating the *DocRoute* and *RecipName* structures within the globals sub stream. Nullify the *rgchSSAddr* attributes from both. Nullify the *ulEISize*

attributes accordingly.

ER.3 Replace: Replace e-mail addresses by isolating the *DocRoute* and *RecipName* structures within the globals sub stream. Modify the *rgchSSAddr* attributes from both. Nullify the *ulEISize* attributes accordingly.

ER.4 External Filter Required: Present any text in the routing slip records to the action that is configured for text.

ER.5 Review: N/A

REFERENCE:
See [MS-XLS], Sections 2.4.91 DocRoute & 2.4.216 RecipName

PRODUCT: POWERPOINT

LOCATION:
The *DocRoutingSlipAtom* contains the attributes *originatorString* and *rgRecipientRoutingSlipStrings*. *originatorString* holds *DocRoutingSlipStrings* and *rgRecipientRoutingSlipStrings* contains an array of *DocRoutingSlipStrings*.

See [MS-PPT].pdf, Sections 2.11.1 & 2.11.2

RECOMMENDATION:

PR.1 Validate: N/A

RP.2 Remove: Remove e-mail addresses by isolating the DocRoutingSlipAtom and nullifying the *DocRoutingSlipStrings* from the *originatorString* and the *rgRecipientRoutingSlipStrings* arrays.

PR.3 Replace: Replace e-mail addresses by isolating the *DocRoutingSlipAtom* and modifying the *DocRoutingSlipStrings* from both the *originatorString* and the *rgRecipientRoutingSlipStrings* arrays.

PR.4 External Filtering Required: Present any text in the routing slip records to the action that is configured for text.

PR.5 Review: N/A

REFERENCE:
See [MS-PPT], Sections 2.11.1 DocRoutingSlipAtom & 2.11.2 DocRoutingSlipString

OFFICE 2003.4.14: END

OFFICE 2003.4.15: Printer Information

DESCRIPTION:
This metadata includes printer information in the document. Often, printer setup information resides in a MS Word or Excel document. In the case of network printers, this information may provide dangerous insight into an enterprise's internal network and less sensitive printer model names.

This item applies to MS Word, and Excel (97 and higher versions).

CONCERN:
The paths and shares provide insight into the networks and paths of the organization, making them a data disclosure threat and a potential invitation to attack.

PRODUCT: WORD

LOCATION:
The *PrDrvr* structure contains numerous records of information about the selected printer. In addition, *PrEnvLand* and *PrEnvPort* structures exist for landscape and portrait information, respectively. They are pulled from the printer as binaries. Although listed as unused, one should probably consider them.

RECOMMENDATION:

WR.1 Validate: N/A

WR.2 Remove: Remove all printer information by deleting the *PrDrvr* record. Adjust the *fcPrDrvr* and *lcbPrDrvr* attributes and effected offsets in *FibRgFcLcb97* accordingly.

Isolate the *PrDrvr* record and nullify the *szPrinter*, *szPrPort*, *szPrDriver*, and *azTruePrnName.* Adjust the *fcPrDrvr* and *lcbPrDrvr* fields and affected offsets in *FibRgFcLcb97* accordingly.

WR.3 Replace: N/A

WR.4 External Filtering Required: Present any text in the printer information fields to the action that is configured for text.

WR.5 Review: N/A

REFERENCE:
See [MS-DOC], Sections 2.9.208 PrDrvr, 2.9.209 PrEnvLand, & 2.9.210 PrEnvPort

PRODUCT: EXCEL

LOCATION:
In Excel, the *Pls* structure contains printer driver and settings information. The *Pls* contains an *rgb* attribute that, in turn, is a *DEVMODE* structure.

RECOMMENDATION:

ER.1 Validate: N/A

ER.2 Remove: Remove printer information by isolating the *Pls* structure within the Dialog Sheet sub stream and nullify the *DEVMODE* structure contained in the *rgb* attribute. Also, delete the printer's private data that follows by *dmDriverExtra* number of bytes after the public portion of *DEVMODE.*

ER.3 Replace: N/A

ER.4 External Filtering Required: Present any text in the printer information to the action that is configured for text.

ER.5 Review: N/A

REFERENCE:

See [MS-XLS], Section 2.4.199 Pls

http://msdn.microsoft.com/en-us/library/dd183565%28VS.85%29.aspx

OFFICE 2003.4.15: END

OFFICE 2003.4.16: Smart Tags

DESCRIPTION:

This metadata includes tags applied to text that match a defined pattern, allowing specific actions to execute based on the category of the smart tag. Smart Tags are an Office feature that enables association of specific actions with text content that matches a pattern associated with each category of Smart Tags. For example, stock ticker symbols can be recognized and tagged to make related actions available to the user whenever the user encounters a ticker symbol.

This item applies to MS Word, PowerPoint, and Excel (97 and higher versions).

CONCERN:

Smart Tags can contain a variety of actions and can be extended with third-party software. Smart Tags should be treated as executable code; therefore, one should consider them a data disclosure and attack threat.

REFERENCE:

See [MS-OSHARED], Section 2.3.4 SmartTag Objects

PRODUCT: WORD

LOCATION:

All Smart Tag data in Word are stored in the *SmartTagData* structure. From *SmartTagData*, records exist for *PropertyBagStore* and an array of *PropertyPags*.

RECOMMENDATION:

WR.1 Validate: N/A

WR.2 Remove: Remove all Smart Tag information by nullifying the *SmartTagData* record. Adjust the *fcFactoidData* and *lcbFactoidData* attributes and effected offsets in *FibRgFcLcb2002* accordingly.

WR.3 Replace: N/A

WR.4 External Filtering Required: Present the text of Smart Tags to the action that is configured for text.

WR.5 Review: N/A

REFERENCE:

See [MS-DOC], Section 2.9.248 SmartTagData

PRODUCT: EXCEL

LOCATION:
In Excel, *Feat* structures store data regarding shared features. A type of *Feat* is *FeatSmartTag*, which contains Smart Tag information.

RECOMMENDATION:

ER.1 Validate: N/A

ER.2 Remove: Remove Smart Tags by scanning for *Feat* in sub-streams and nullifying any *FeatSmartTag* records.

E**R.3 Replace**: Replace Smart Tags by scanning for *Feat* in sub-streams and modifying any *FeatSmartTag* records.

ER.4 External Filtering Required: N/A

ER.5 Review: N/A

REFERENCE:
See [MS-XLS], Sections 2.5.110 FactoidData, 2.5.125 FeatSmartTag, 2.5.134 FrtFlags, 2.5.135 FrtHeader, & 2.4.111 Feat

PRODUCT: POWERPOINT

LOCATION:
In PowerPoint, a *PP11DocBinaryTagExtension* structure can contain a *SmartTagStore11Container*. The SmartTagStore11Container contains Smart Tag information.

RECOMMENDATION:

PR.1 Validate: N/A

PR.2 Remove: Remove Smart Tags by isolating the *PP11DocBinaryTagExtension*, if present, within the *DocProgTagsContainer* and nullifying the *DocProgTagsContainer*.

PR.3 Replace: N/A

PR.4 External Filtering Required: Present any text in Smart Tags to the action that is configured for text.

PR.5 Review: N/A

REFERENCE:
See [MS-PPT], Sections 2.4.23.7 PP11DocBinaryTagExtension, & 2.11.28 SmartTagStore11Container

OFFICE 2003.4.16: END

OFFICE 2003.4.17: Meeting Minder

DESCRIPTION:

This item includes meeting minutes entered via the PowerPoint Meeting Minder feature. Meeting minutes can attach to PowerPoint documents with the PowerPoint Meeting Minder feature and typically are associated with an action item list. The action item list is included in the presentation as part of a slide or series of slides. The associated minutes are accessible only through the Meeting Minder user interface.

This issue applies to MS PowerPoint 97, but not the 2003 version.

CONCERN:

Any free-form text field provides potential data disclosure threats. Consider Meeting Minder in PowerPoint 97 documents a data disclosure threat.

RECOMMENDATION

PR.1 Validate: N/A

PR.2 Remove: Remove meeting minder data.

PR.3 Replace: N/A

PR.4 External Filtering Required: Present any text in the record to the action that is configured for text.

PR.5 Review: Present the meeting minder text to the human reviewer.

OFFICE 2003.4.17: END

OFFICE 2003.4.18: Image Properties

DESCRIPTION:

Images in Office applications can be manipulated to affect their display characteristics, such as resizing and cropping. When this manipulation occurs, the content of the original image persists within the document as the viewing application renders the changes. For example, when the size of a picture is modified in Word, the full-size image resides inside the document and Word performs the image resizing when displayed. This is also true for most of the image manipulation tools that can be applied in Word, Excel, and PowerPoint.

CONCERN:

Risks include all images in the document, including their alternate text, description, source, and hyperlink (if present). One should identify when images are grouped, cropped, placed off the page border, or reduced in size by more than 50%, or when brightness or contrast is adjusted (and how much). Also, identify if images reside in headers or footers, are hidden, are used as watermarks or document thumbnails, or are outside the default viewable area of Excel documents. Identify when

images have display filters or adjustments, such as shape, transparency, blur, and angle.

Where feasible, identify when images (or other objects) overlap or conceal text or other objects. This is technically complex, as images can be identified with either relative coordinates or absolute coordinates. When absolute coordinates are used, the object's location can be easily identified and the ability to calculate overlapping objects is relatively simple. The complexity increases when relative coordinates and/or text wrapping are used to determine object locations. Because the object's displayed location relies heavily on the rendering engine of the application, the exact locations of the objects are difficult to identify. In some cases, this may make some of the alternative recommendations below extremely technically complex to achieve.

PRODUCT: WORD, EXCEL, POWERPOINT

LOCATION:
Image manipulation properties, called *OfficeArtFOPTEOPID* records, and their corresponding values, op, are stored in record called *OfficeArtFOPTE*. Groupings of these key-value pairs are held in records called *OfficeArtRGFOPTE*.

OfficeArtRGFOPTE primarily uses two important structures. First, it exists in the drawing group record, called *OfficeArtDggContainer*, which holds all Binary Large Image or Picture (BLIP) data (images) for the document. When inside this main drawing group record, the *OfficeArtRGFOPTE* table describes the default property values for all images in the document. Inside the *OfficeArtDggContainer*, the default property tables are referred to by the *OfficeArtFOPT* and *OfficeArtTertiaryFOPT* records. Each of these records holds an *OfficeArtRGFOPTE* property table.

The *OfficeArtRGFOPTE* property table is also used for the second structure, *OfficeArtSpContainer* record, which describes the properties for individual images and shapes. Fewer records exist in these property tables, as they exist only to describe the difference between the individual image properties and the default properties for images in the document. In the *OfficeArtSpContainer*, the *OfficeArtRGFOPTE* records reside in three property blocks--*OfficeArtFOPT*, *OfficeArtSecondaryFOPT*, and *OfficeArtTertiaryFOPT*.

RECOMMENDATION:
For Grouped Images:
AR.1.1 Validate: N/A

AR.1.2 Remove: Remove grouped images and adjust the *OfficeArtFOPTEOPID, OfficeArtFOPTE, and OfficeArtRGFOPTE* data structures accordingly.

AR.1.3 Replace: Replace grouped images by combining grouped parts into a single flattened image. Discard any non-visible data and adjust *OfficeArtFOPTEOPID, OfficeArtFOPTE,* and *OfficeArtRGFOPTE* data structures accordingly.

AR.1.4 External Filtering Required: Pass all images to the action that is configured for handling images of that type.

AR.1.5 Review: Pass all images for review.

For Resized Images:
AR.2.1 Validate: N/A

AR.2.2 Remove: Remove cropped images that exceed a configured threshold for the amount of permissible image cropping and adjust the *OfficeArtFOPTEOPID, OfficeArtFOPTE, and*

OfficeArtRGFOPTE data structures accordingly.

AR.2.3 Replace: Replace the original image by applying the cropping to the image, storing just the cropped image in the file, discarding the cropping parameters, and adjusting the *OfficeArtFOPTEOPID, OfficeArtFOPTE, and OfficeArtRGFOPTE* data structures accordingly.

AR.2.4 External Filtering Required: Pass the cropped image to the action configured for that image type.

AR.2.5 Review: Pass the cropped image for review.

For Brightness- and Contrast- Adjusted Images:
AR.3.1 Validate: N/A

AR.3.2 Remove: Remove images with brightness or contrast properties that fall outside of configured limits for brightness, and contrast and adjust the *OfficeArtFOPTEOPID, OfficeArtFOPTE, and OfficeArtRGFOPTE* data structures accordingly.

AR.3.3 Replace: Replace the original image by applying the adjustment to the image, removing the brightness or contrast settings from the file, and adjusting the *OfficeArtFOPTEOPID, OfficeArtFOPTE, and OfficeArtRGFOPTE* data structures accordingly.

AR.3.4 External Filtering Required: Pass the adjusted image to the action that is configured for that image type.

AR.3.5 Review: Pass the adjusted image for review.

Resized Images
AR.4.1 Validate: N/A

AR.4.2 Remove: Remove resized images that exceed a configured threshold for resizing and adjust the *OfficeArtFOPTEOPID, OfficeArtFOPTE, and OfficeArtRGFOPTE* data structures accordingly.

AR.4.3 Replace: Replace the original image by applying resizing to the picture, discarding the size setting from the file, and adjusting the *OfficeArtFOPTEOPID, OfficeArtFOPTE, and OfficeArtRGFOPTE* data structures accordingly.

AR.4.4 External Filtering Required: Pass the resized image to the action that is configured for that image type.

AR.4.5 Review: Pass the resized image for review.

Overlapping Images
AR.5.1 Validate: N/A

AR.5.2 Remove: Remove the overlapped images by using the image attributes to detect overlap. Delete all involved images and adjust the *OfficeArtFOPTEOPID, OfficeArtFOPTE, and OfficeArtRGFOPTE* data structures accordingly.

AR.5.3 Replace: Replace overlapped images by using the image attributes to detect overlapped images, grouping the involved images into a single image, and adjusting the *OfficeArtFOPTEOPID, OfficeArtFOPTE, and OfficeArtRGFOPTE* data structures accordingly.

AR.5.4 External Filtering Required: Pass the images to the action that is configured for images of that type.

AR.5.5 Review: Pass all images for review.

Document-Specific Image Metadata
AR.6.1 Validate: N/A

AR.6.2 Remove: Remove alternate text, description, or source data.

AR.6.3 Replace: N/A

AR.6.4 External Filtering Required: Pass alternate text, description, or source data to the action that is configured for text.

AR.6.5 Review: N/A

REFERENCE:
See "Guidelines for Microsoft Office OpenXML File Analysis and Sanitization Tools."

PRODUCT: WORD

LOCATION:
In MS Word, the *OfficeArtDggContainer* resides in the *OfficeArtContent* record, along with an array of *OfficeArtWordDrawing* records. Each *OfficeArtWordDrawing* record identifies a specific blip in the *OfficeArtDggContainer's* collection of blips with the *dgglbl* value. The *OfficeArtWordDrawing* also stores an *OfficeArtDgContainer*, which holds an *OfficeArtSpContainer*. With the *OfficeArtDggContainer* and *OfficeArtSpContainer* records, the property tables can be located and modified.

Separate from the image properties in the property tables, the location of the image and the word wrapping settings are stored in a record called *Spa*. The *Spa* record resides in the *PlcfSpa* record, which identifies all the Spas for the document.

RECOMMENDATION:
Images Off of the Page, Borderpage border
WR.1.1 Validate: N/A

WR.1.2 Remove: Remove images that are off of the page border by using the image attributes to detect this condition, and deleting the images and adjusting the attributes in *OfficeArtContent* according. If an image is partially off of the page border, remove the portion of the image that is off of the border.

WR.1.3 Replace: N/A

WR.1.4 External Filtering Required: Pass images that are partially or fully outside of page borders to the action that is configured for that image type.

WR.1.5 Review: Pass image partially or fully outside of page boundaries for review.

Images used as watermarks
WR.2.1 Validate: NA

WR.2.2 Remove: Remove watermark image by deleting the image and adjusting the attributes in *OfficeArtContent* accordingly.

WR.2.3 Replace: N/A

WR.2.4 External Filtering Required: Pass the watermark image to the action that is configured for that image type.

WR.2.5 Review: Pass the image for reviews.

PRODUCT: EXCEL

LOCATION:
The *OfficeArtDggContainer* and *OfficeArtDgContainer* records reside in the *HFPicture* records. The *HFPicture* records can be found within the *WORKSHEETCONTENT*, *CHARTSHEETCONTENT*, *DIALOGSHEETCONTENT*, *WORKBOOKCONTENT*, and *MACROBOOKCONTENT* attributes. The *OfficeArtDgContainer* holds an *OfficeArtSpContainer,* which holds image property tables. The *OfficeArtDggContainer* holds image property tables also. With the *OfficeArtDggContainer* and *OfficeArtSpContainer* records, the property tables can be located and modified.

RECOMMENDATION:
Images outside the Default Viewable Area of an Excel File
ER.1.1 Validate: N/A

ER.1.2 Remove: Remove images in extreme locations, such as images that are far from non-empty cells or in a non-default viewable area, and adjust the fields in *OfficeArtDggContainer* and *OfficeArtDgContainer.*

ER.1.3 Replace: N/A

ER.1.4 External Filtering Required: Pass images to the action that is configured for that image type.

ER.1.5 Review: Pass the image for review.

PRODUCT: POWERPOINT

LOCATION:
In MS PowerPoint, the *OfficeArtDggContainer* resides in the *DrawingGroupContainer* record, which exists in the *DocumentContainer*. The *OfficeArtDgContainer*, which holds an *OfficeArtSpContainer*, is stored in the *DrawingContainer* record. The *DrawingCotainer* record can reside in the *HandoutContainer*, *MainMasterContainer*, *NotesContainer* and *SlideContainer* attributes. With the *OfficeArtDggContainer* and *OfficeArtSpContainer* records, the property tables can be located and modified.

RECOMMENDATION:
Images Off of the Page, Borderpage border
PR.1.1 Validate: N/A

PR.1.2 Remove: Remove images that are off of the page border and adjust the attributes in *OfficeArtDggContainer* accordingly. If an image is partially off of the page border, remove the portion of the image that is off of the border.

PR.1.3 Replace: N/A

PR.1.4 External Filtering Required: Pass the image to the action that is configured for the image type.

PR.1.5 Review: Pass the image for review.

OFFICE 2003.4.18: END

OFFICE 2003.4.19: Windows Metafile (WMF) and Enhanced Metafile (EMF)

DESCRIPTION:

WMF images are an ordered sequence of drawing instructions, or commands, that produce an image. A WMF file may contain vector, text, or bitmap objects to display. They are application-independent so that images can be shared among applications, but they are the native formats for Office and MS clipart.

EMF is an extension of WMF that adds support to additional commands for 32-bit versions of Windows. The EMF format contains a superset of the 16-bit WMF format commands.

CONCERN:

WMF and EMF files potentially can pose an attack risk because the files have the ability to include executable code and automatic execute upon opening. Microsoft fixed this currently known exploit with a patch release on supported OSs. This vulnerability still applies to unpatched and unsupported Windows versions.

These formats are also susceptible to data disclosure because the metafile format is a container for text and bitmap objects, which can potentially contain sensitive data. The distinct formats within the metafile format, such as text and bitmap, should require separate filter processing.

One way to reduce the threat of hidden data in images is to flatten them and remove any layering, hidden data, and metadata. During flattening, images may reduce in size to save space and reduce the data amount in each image without significantly impacting the Office document appearance, as long as the size does not reduce to a value smaller than the visible area of the image in the file. The ability to modify the image without loss also depends upon the type of image used, as compressed formats, such as jpeg, are more prone to loss during conversion than other lossless image types.

Image flattening can result in a variety of different target image formats, which potentially include PNG or JPEG. Although PNG is an increasingly attractive option because of the balance between file size and quality, a single file type may not exist to balance the quality and size issues for a variety of source image types and characteristics. Where the original image can be flattened reliably, the original image type could be maintained also. The goal of this document is not to require a specific target image format. It is to communicate the importance of flattening images in a way that balances size and quality for users.

The range of image types carries varying risk levels. Vector images and drawings may contain additional information, such as metadata and text. This information may remain unapparent to the

user, even if the user visually inspects the image carefully. Vector graphics define shapes and content with mathematical equations. The most significant difference between this and raster images, which represent data as an array of pixels, is that one can scale vector images without the quality loss that typically results when scaling raster images. The retention of additional information about the image content supports the ability to provide enhanced quality when scaling. Therefore, vector images may carry additional risk. Flattened raster image formats, with removed layers, are "what you see is what you get." Some common vector formats (although they may also include raster content) include WMF, EMF, and the new Scalable Vector Graphics (SVG) format. Some examples of raster formats include JPEG, GIF, BMP, and TIFF.

The conversion of vector image formats to flattened raster formats has benefits and drawbacks. One usually can search for text within a vector image, whereas one cannot search text within a raster image unless using Optical Character Recognition (OCR). Therefore, flattening a vector image eliminates text search capability while reducing the ability to enlarge the image in the future without introducing graininess. Benefits counter these drawbacks, including the ability to prevent data hiding behind images, or text hidden within an image that a user cannot see when looking at the image.

See "Guidelines for Microsoft Office OpenXML File Analysis and Sanitization Tools."

PRODUCT: WORD, EXCEL, POWERPOINT

LOCATION:

Word, Excel, and PowerPoint applications share the container element, *OfficeArtDggContainer*. It stores an array of image formats, including WMF and EMF images. Each application implements the container differently.

The *OfficeArtDggContainer* holds an *OfficeArtBStoreContainer* record that specifies the container for all BLIP objects in the document. The variable name blipStore in the *OfficeArtDggContainer* references the *OfficeArtBStoreContainer*.

The *OfficeArtBStoreContainer* holds an *OfficeArtRecordHeader* record and an array of *OfficeArtBStoreContainerFileBlock* records. The *OfficeArtRecordHeader*, identified by the variable name *rh*, holds subfields that describe the second record in the container, *OfficeArtBStoreContainerFileBlock*, identified by rgfb.

Either *OfficeArtFBSE* or *OfficeArtBlip* records can make up the O*fficeArtBStoreContainerFileBlock* array. The parent element, *OfficeArtBStoreContainer*'s *OfficeArtRecordHeader*, identifies which type of records exists in the array.

The *OfficeArtBlip* field is a container for one of many types of BLIP types. The *OfficeArtBlip* field in the *OfficeArtBStoreContainerFileBlock* array holds *OfficeArtBlipWMF* and *OfficeArtBlipEMF* records for WMF and EMF files, respectively.

The *OfficeArtBlipWMF* and *OfficeArtBlipEMF* records have an identical structure, but they differ in the record values. Both have the *OfficeArtRecordHeader*, *rgbUid1*, *rgbUid2*, *OfficeArtMetafileHeader*, and *BlipFileData* records. The actual EMF and WMF data exists in the variable length field *BlipFileData*.

RECOMMENDATION:

AR.1 Validate: N/A

AR.2 Remove: Remove WMF and EMF files by removing *OfficeArtBlipWMF* and *OfficeArtBlipEMF* records and adjusting the pointers in the parent objects accordingly.

AR.3 Replace: Replace WMF and EMF files with flattened raster images by converting the bytes in the *BlipFileData* fields of *OfficeArtBlipWMF* and *OfficeArtBlipEMF* records to raster images, adjusting the pointers in the parent objects accordingly, and storing the converted files in the document.

AR.4 External Filtering Required: Present the WMF or EMF file to the action that is configured for that file type.

AR.5 Review: Present the WMF or EMF images for review.

PRODUCT: WORD

LOCATION:
Word stores the *OfficeArtDggContainer* record within the *OfficeArtContent* record. The variable name *DrawingGroupData* identifies it. The *OfficeArtContent* record exists in the *WordDocument Stream*. Because the *WordDocument Stream* has no predefined structure, the *OfficeArtContent* record can exist anywhere in the stream, other than directly at the beginning where the FIB exists.

PRODUCT: EXCEL

LOCATION:
In Excel, the *OfficeArtDggContainer* can exist as part of the main document or as part of the sheet header or footer.

If the container is part of the main document, the *OfficeArtDggContainer* exists in the record *MsoDrawingGroup* and is identified by the variable name *rgChildRec*. The *MsoDrawingGroup* record, followed by a *Continue* record or multiple *Continue* records, makes up a complete *MSODRAWINGGROUP* record. A *Continue* record specifies a continuation of data in the preceding *MsoDrawingGroup* record. If records exist with data that exceeds 8,224 bytes in length, the record must be split into several *Continue* records. The *MSODRAWINGGROUP* record can exist from zero to none, one or many times in the *WORKBOOKCONTENT* stream.

If the container exists as part of the sheet header or footer, the *OfficeArtDggContainer* exists in the *HFPicture* record and is identified by the rgDrawing variable. This *HFPicture* record can exist in several sub-streams in an Excel document, including in the *CHARTSHEETCONTENT*, *MACROSHEETCONTENT*, *WORKSHEETCONTENT*, *DIALOGSHEETCONTENT* and *WORKBOOKCONTENT* streams.

PRODUCT: POWERPOINT

LOCATION:
In PowerPoint files, the *OfficeArtDggContainer* record resides in the *DrawingGroupContainer* record, which is identified by the variable name *OfficeArtDgg*. The *DrawingGroupContainer* resides in the *DocumentContainer* record, which exists as a top-level record in the *PowerPoint Document Stream*.

OFFICE 2003.4.19: END

OFFICE 2003.4.20: Database Connections and Queries

DESCRIPTION:

Office supports powerful connectivity to databases, which results in database connection and query information storage in Office documents. Word uses database connections to drive the Mail Merge capabilities, which includes generating mailing labels from an address database. Excel provides extensive support for pulling data from external data sources, including databases and OLAP cubes. It can display data in tables, graphs, or PivotTables.

CONCERN:

Sensitive information, that could result in data disclosure if not properly filtered, includes a path or URL to the database file or database server, the database username, password, and Structured Query Language (SQL) query string. The database file path or database server address potentially could reveal information about the internal organization's setup. Usernames and passwords provide malicious users with additional information to assist in an attack. SQL queries could provide outsiders with a view of the internal database structure and reveal sensitive table names, column names, and filtering criteria.

PRODUCT: WORD

LOCATION:

Database connections within a Word document can be created in two ways; through an included field or through the Mail Merge function. For information relating to the structure of field codes, see OFFICE 2003.3.1 Field Codes.

In MS Word, Print and Mail Merge documents use a complex structure called the *Pms*. The *Pms* holds the connection information to a data source, such as a database, and query information, such as a SQL string. The *Pms* resides in the Table Stream. An index and offset, or length, held in the *FibRgFcLcb97* or *FibRgFcLcb2002* record, can locate it.

The *FibRgFcLcb97* and *FibRgFcLcb2002* records contain integers, *fcPms* and *fcPmsNew* respectively, which denote the location of the *Pms* in the Table Stream. It also has the integers, *lcbPms* and *lcbPmsNew*, which indicate the size of the *Pms* structure by the offset from the location value.

Additionally, the *FibRgFcLcb2002* record contains *fcODSO* and *lcbODSO* records that indentify the location and offset of an Office Data Source Object (ODSO). The ODSO contains data to perform a mail merge. It stores data in an array of *ODSOPropertyBase items*.

Inside the *Pms* record, the important fields include *lxszSqlStr* and *sttbfRfs*. The *lxszSqlStr* is the SQL Query string in Unicode that executes against the database connection. The next is *sttbfRfs*, a string table that contains the strings for a mail merge connection.

Note: Remove database connection information, including usernames, passwords, and database locations, from Word and Excel files to protect the organization's privacy. Removing this sensitive information will not affect the existing data that is already stored in the Excel or Word document. After removing sensitive information, users should not attempt to refresh the information from the source data, as the broken connections can cause issues. Remove connection credentials, so that the username and password are removed from the connections.

Removing query strings ensures non-exposure of the internal database structures, even partially.

Although SQL queries alone do not provide an exploit into the system, the table names, columns or search criteria may expose sensitive information outside the organization.

Removing the query strings is also important for ensuring that the internal database structures are not exposed, even partially. Although the SQL queries alone do not provide an exploit into the system, the table names, columns, or search criteria may expose sensitive information outside the organization

RECOMMENDATION:

WR.1 Validate: Ensure referential integrity and consistency in *Pms*, *FibRgFcLcb97* and *FibRgFcLcb2002* records by tracing the index and size attributes related to print and mail merge, and verifying the location and size of the actual query.

WR.2 Remove: Remove connection strings by tracing the index and size attributes that are related to queries in the *Pms*, *FibRgFcLcb97* and *FibRgFcLcb2002* records, and deleting or nullifying the queries.

WR.3 Replace: N/A

WR.4 External Filtering Required: Pass text data found in the queries to the action that is configured for text.

WR.5 Review: N/A

PRODUCT: EXCEL

LOCATION:

Excel uses zero or many *DConn* records that exist in the *WORKBOOKCONTENT* sub-stream. Each *DConn* record specifies information for a single data connection. This *DConn* record can represent one of many types of data sources: Open Database Connectivity (ODBC)-based, Data Access Object (DAO)-based, Web query, OLE DB-based, text-based created via text query, and ActiveX Data Objects (ADO) record set.

The notable fields in the *DConn* record include the connection field, the *rgbSQL*, and the *rgbSQLSav* fields. The connection field can hold either a connection string, a *DConnConnectionWeb* object for connection information for a Web query, a *DConnConnectionOleDb* object for connection information for an OLE database (DB) connection string, or a *TxtQry* for information for a text query. This is where database path, server locations, username, and passwords may persist. The *rgbSQL* and *rgbSQLSav* fields contain actual SQL statements to execute against the data source.

Excel also uses the *DbQuery* structure. It stores query information and is necessary for displaying external data in either a table or a PivotTable. Data in a PivotTable can be cached with the *PivotCache* record. If the *PivotCache* record is used with data from an external data source, then the *DbQuery* records will exist in an *SXSRC* record, which is stored in the *PIVOTCACHEDEFINITION*. The *PIVOTCACHEDEFINITION* also describes the type of data source with its *SXVS* record (a value of 0x002 represents an External Source and *DbQuery* will exist inside the *SXSRC* record). Zero or more *PIVOTCACHEDEFINITIONs* may reside inside the *WORKBOOKCONTENT* sub-stream along with the previously mentioned *DConn* record.

The *DbQuery* record is also used within the *QueryTable* record. The *DbQuery* and *DBQueryExt* records constitute the bulk of the *QueryTable* record. Multiple *QueryTable* records can reside in the WORKSHEETCONTENT sub-stream.

In addition to the *DbQuery* and *DBQueryExt* records, an *SXADDLDBQUERY* record is used for

additional connection information for PivotTable views, *PivotCaches*, or query tables. These records can contain connection and SQL query strings. The *SXADDLDBQUERY* resides in the *SXADDLCACHE* record (an optional record in the *PIVOTCACHEDEFINITION* record), and in the *SXADDLQSI* record, an optional record in the QueryTable record.

RECOMMENDATION:

ER.1 Validate: Ensure consistency in the *DConn* and *DbQuery* but verifying the correctness (size and location) of the sub-structures holding the connection and query information.

ER.2 Remove: Remove connection strings by tracing the attributes and sub-structures within the *DConn* and *DbQuery* records and nullifying or deleting all connection information in the records.

ER.3 Remove: Remove query strings by tracing the attributes and sub-structures within the *DConn* and *DbQuery* records and nullifying or deleting all query information in the records.

ER.4 Replace: N/A

ER.4 External Filtering Required: Pass all text in the query connection to the action that is configured for text.

ER.5 Review: N/A

REFERENCE:
See [MS-XLS].pdf, Sections 2.2.8.3 through 2.2.8.8, for specific details on how DbQuery, DBQueryExt, and DConn records differ for the different data source connections (e.g., OLE DB, ODBC).

OFFICE 2003.4.20: END

The next three constructs (Tracked Changes, Versions, and Fast Save Data) are interrelated features of MS Office 2003 documents. MS specifications do not document data structures related to these constructs clearly, so location information is missing and recommendations are general for these constructs. Properly following the recommendations will require additional research on the construct by the implementer.

OFFICE 2003.4.21: Tracked Changes

DESCRIPTION:
This item includes all tracked changes in the document. Office's change tracking feature tracks insertions, deletions, and formatting changes that users make to the document. Such changes contain deleted text and author and date information that may unintentionally remain in the document upon distribution. This item applies to MS Word and Excel (97 and higher versions).

CONCERN:

Tracked changes can include user names, dates of changes, and old content. As a result, one should consider tracked changes a data disclosure concern. Consider the possibility that prior changes could also include other metadata items. This consideration opens up the possibility that tracked changes could inherit many of the other metadata concerns.

RECOMMENDATION:

AR.1 Validate: N/A

AR.2 Remove: Remove all tracked change data.

AR.2 Remove: Remove user names from tracked changes.

AR.2 Remove: Remove dates from tracked changes.

AR.3 Replace: N/A

AR.4 External Filtering Required: Pass all text in the tracked changes to the action that is configured for text data.

AR.5 Review: Present final document for review.

OFFICE 2003.4.21: END

OFFICE 2003.4.22: Versions

DESCRIPTION:

This item includes version information in Word documents. The versioning feature (File -> Versions) in Word allows one to save multiple historical versions of a document within a single file. Versioning has usefulness during document creation, but it is potentially sensitive after document release.

This issue applies to MS Word 97 and higher versions.

CONCERN:

Prior Word versions contain potential data disclosure concerns. Consider that prior versions could also include other metadata items. This consideration provides the possibility that prior versions could inherit many of the other metadata concerns.

RECOMMENDATION:

WR.1 Validate: N/A

WR.2 Remove: Remove version information from document.

WR.3 Replace: N/A

WR.4 External Filtering Required: N/A

WR.5 Review: Present all version information for review.

OFFICE 2003.4.22: END

OFFICE 2003.4.23: FastSave Data

DESCRIPTION:

Fast Save allows quicker file saves by tracking and only storing changes since the last save. This results in a file, often larger than normal, because it can carry information from previous revisions. From the *Tools* menu, select *Options* and click on the *Save* tab. The Fast Save feature is generally available as a checkbox control. This issue applies to MS Word, PowerPoint, and Excel (97 and higher versions). Office Service Pack 3 disables FastSave for security reasons.

CONCERN:

Microsoft disabled FastSave via Office 2003 Service Pack 3, citing security concerns. Prior content versions, including metadata, may remain in the document due to using FastSave. The use of FastSave presents the possibility that prior versions could inherit many of the other metadata concerns.

RECOMMENDATION:

AR.1 Validate: N/A

AR.2 Remove: Remove all *Fast Save* information.

AR.3 Replace: N/A

AR.4 External Filtering Required: Pass text in FastSave data to the action that is configured for text.

AR.5 Review: N/A

OFFICE 2003.4.23: END

OFFICE 2003.4.24: Document Protections

DESCRIPTION

Document protections refer to any security features to apply to documents to protect content from viewing or modification. Forms of this protection include simple write protection, where passwords may be stored in clear-text; read protection, where the document may be obfuscated or encrypted; or digital signatures to ensure that content remains unaltered.

CONCERN

Write Reservation Password:

With write protection, where a password is required to edit an Office document, passwords reside in clear-text (for Word and PowerPoint formats). Although write-protection exists to prevent accidental editing, this field accidentally may expose sensitive passwords.

Even Excel's use of the Password Verifier Algorithm for locking parts of the workbook from editing has security holes that reveal the original password.

Encryption:

Documents can be password- protected from viewing via Exclusive Or (XOR) obfuscation, Office binary document RC4 encryption, or Office binary document RC4 CryptoAPI encryption. When XOR obfuscation alone protects Word documents, one easily can extract data and retrieve the document password, which could result in a data disclosure risk.

For documents that contain encryption and obfuscation, certain data persists outside of the encryption and could pose a threat for data disclosure. In Word documents, the *ObjectPool* storage, *Macros* storage, *Custom XML Data* storage, *XML Signatures* storage, and *Signatures* stream are not obfuscated or encrypted. Additionally, the *Document Summary Information* and *Summary Information* streams are not obfuscated or encrypted if the *fDocProps* property is set to false in the *EncryptionHeader.Flags* record.

When using Office binary document RC4 encryption or Office binary document RC4 CryptoAPI encryption, the *WordDocument* stream, the *Table* stream, and the entire *Data* stream reuse the same block numbers. This reuse can occur potentially with known clear-text, implying that one can directly extract or retrieve certain portions of encrypted data.

During encryption, Excel encodes only certain storages and streams. For encrypted records, Excel does not encrypt the record type and size in BIFF streams. Therefore, one can read the list of records present in the file without decrypting.

Digital Signatures:

In addition to the security issues associated with password protecting documents, digital signatures do not provide strong enough protection to prevent a user from sneaking data inside the document. Certain streams and storages are not subject to signing. One can modify these streams or storages without invalidating the signature.

PRODUCT: WORD

LOCATION

Write Reservation Password:

The write-reservation password is embedded in clear text in Word documents. The flag *FibBase.fWriteReservation* specifies if the document has a write-reservation password. If it does, then the *SttbfAssoc* record will contain the password in clear text at index 0x11. Note that the password must contain 15 characters or less.

Encryption:

A file in Word Binary File Format can be password protected using three different mechanisms: XOR obfuscation, Office binary document RC4 encryption, and Office binary document RC4 CryptoAPI encryption.

If *FibBase.fEncryption* and *FibBase.fObfuscation* are both 1, then the file is obfuscated using XOR obfuscation, as specified in Section 2.2.6.1.

If *FibBase.fEncryption* is 1 and *FibBase.fObfuscation* is 0, the file is encrypted using either Office Binary Document RC4 Encryption or Office Binary Document RC4 CryptoAPI Encryption, as specified in Sections 2.2.6.2 and 2.2.6.3, with the *EncryptionHeader* stored in the first *FibBase.lKey* bytes of the Table stream. The *EncryptionHeader.EncryptionVersionInfo* specifies the encryption mechanism that encrypted the file.

The encrypted document data is stored in the optional stream, and the name must be "encryption." This stream must not be present unless the document is encrypted with Office Binary Document RC4 CryptoAPI Encryption and *fDocProps* is set in the *EncryptionHeader.Flags*.

See [MS-DOC].pdf, Section 2.2.6, Encryption and Obfuscation (Password to Open).

Digital Signatures:

Signatures in a Word document reside optionally in either the Signatures stream, identified by the name *_signatures*, or in the XML Signatures Storage identified by the *_xmlsignature*s storage. The *_signatures* stream must contain exactly one CryptoAPI Digital Signature structure. The CryptoAPI Digital Signature structure must contain at least one CryptoAPI Digital Signature *CertificateInfo*

structure.

Signatures stored in the *_xmlsignatures* storage must be stored as streams with no header and as UTF-8 characters. The contents of each stream MUST be a valid signature, as specified by [XMLDSig], and generated, as specified in [MS-OFFCRYPTO].pdf, Section 2.5.2. More than one signature can reside in the *_xmlsignatures* storage.

RECOMMENDATION
Write Reservation Password:
WR.1.1 Validate: N/A

WR.1.2 Remove: Remove write- reservation password fields by deleting or nullifying the password filed in SttbfAssoc and setting FibBase.fWriteReservation to indicate the file is not write reservation protected.

WR.1.3 Replace: N/A

WR.1.4 External Filtering Required: N/A

WR.1.5 Review: N/A

Encryption:
WR.2.1 Validate: N/A

WR.2.2 Remove: N/A

WR.2.3 Replace: N/A

WR.2.4 External Filtering Required: N/A

WR.2.5 Review: N/A.

WR2.6 Reject: Reject files containing any encrypted streams or storage.

Digital Signatures:
WR.3.1 Validate: Ensure that the digital signature is correct.

WR.3.2 Remove: Remove storages and streams from documents that are not validated by the digital signature process.

WR.3.3 Replace: Remove the digital signature from the document

WR.3.4 External Filtering Required: Pass the file to a third party application to validate the signature.

WR.3.5 Review: N/A

WR.3.6 Reject: Reject any file containing digital signatures.

PRODUCT: EXCEL

LOCATION
Write Reservation Password:

The *Protect* and *Password* records specify the protection state for the sheet or workbook.

If these records exist in the globals sub-stream, the specified protection state and password apply to the workbook. If these records exist in a worksheet sub-stream, chart sheet sub-stream, macro

sheet sub-stream, or dialog sheet sub-stream, the specified protection state and password apply to only that sheet.

Encryption:

If a file in this format is saved with encryption, it MUST be saved with XOR obfuscation, or RC4 encryption, or one of a number of RC4 CryptoAPI encryption algorithms. The *FilePass* record specifies the specific obfuscation or encryption method used, and the associated obfuscation or encryption information.

If using RC4 CryptoAPI encryption, certain storages and streams reside in the Encryption Stream called named "encryption".

Digital Signatures:

Signatures in an Excel document are stored in either the Signatures stream, identified by the name *_signatures*, or in the XML Signatures Storage, as identified by the *_xmlsignatures* storage. The *_signatures* stream must contain exactly one CryptoAPI Digital Signature structure. The CryptoAPI Digital Signature structure must contain at least one CryptoAPI Digital Signature *CertificateInfo* structure.

Signatures stored in the *_xmlsignatures* storage must reside as streams with no header and as UTF-8 characters. The content of each stream MUST be a valid signature, as specified by [XMLDSig], and generated as specified in [MS-OFFCRYPTO].pdf, Section 2.5.2. More than one signature can reside in the *_xmlsignatures* storage.

Note that signature generation in Excel ignores data in the record data of the *WriteAccess* record in the *Globals* sub stream.

RECOMMENDATION
Write Reservation Password:
ER.1.1 Validate: N/A

ER.1.2 Remove: Remove write- reservation password fields by deleting or nullifying the *Protect* and *Password* records in all sub-streams.

ER.1.3 Replace: N/A

ER.1.4 External Filtering Required: N/A

ER.1.5 Review: N/A

Encryption:
ER.2.1 Validate: N/A

ER.2.2 Remove: N/A

ER.2.3 Replace: N/A

ER.2.4 External Filtering Required: N/A

ER.2.5 Review: N/A

ER2.6 Reject: Reject files containing any encrypted streams or storage.

Digital Signatures:
ER.3.1 Validate: Ensure that the digital signature is correct.

ER.3.2 Remove: Remove storages and streams from documents that the digital signature process

did not validate.

ER.3.3 Replace: Remove the digital signature from the document.

ER.3.4 External Filtering Required: Pass the file to a third party application to validate the signature.

ER.3.5 Review: N/A

ER.3.6 Reject: Reject any file containing digital signatures.

PRODUCT: POWERPOINT

LOCATION

Write Reservation Password:

An application only grants Modify access to the presentation if a user- provided password matches the *modifyPassword* field within the *ModifyPasswordAtom* record. The password resides in clear-text in this field within this field.

Encryption:

The *CurrentUserStream* holds an unsigned integer that specifies a token to identify if the file is encrypted. If the document is encrypted, the *UserEditAtom* contains a *PersistIdRef* that specifies the value to look up in the Persist Object directory to find the offset of the *CryptSession10Container* record, a container record that specifies encryption properties for the file.

If the document is encrypted, the stream named *EncryptedSummary* exists. The *EncryptedSummary* stream contains one or more encrypted summary streams using RC4 CryptoAPI encryption.

Digital Signatures:

Signatures in a PowerPoint document reside in either the Signatures stream, identified by the name *_signatures*, or in the XML Signatures Storage, as identified by the *_xmlsignatures* storage. The *_signatures* stream must contain exactly one CryptoAPI Digital Signature structure. The CryptoAPI Digital Signature structure must contain at least one CryptoAPI Digital Signature *CertificateInfo* structure.

Signatures stored in the *_xmlsignatures* storage must reside as streams with no header and as UTF-8 characters. The contents of each stream MUST be a valid signature, as specified by [XMLDSig], and generated as specified in [MS-OFFCRYPTO].pdf, Section 2.5.2. More than one signature can be present in the *_xmlsignatures* storage.

RECOMMENDATION

Write Reservation Password:
PR.1.1 Validate: N/A

PR.1.2 Remove: Remove write- reservation password fields by deleting or nullifying the *modifyPassword* filed in *ModifyPasswordAtom*.

PR.1.3 Replace: N/A

PR.1.4 External Filtering Required: N/A

PR.1.5 Review: N/A

Encryption:
PR.2.1 Validate: N/A

PR.2.2 Remove: N/A

PR.2.3 Replace: N/A

PR.2.4 External Filtering Required: N/A

PR.2.5 Review: N/A

PR.2.6 Reject: Reject files containing any encrypted streams or storage.

Digital Signatures:
PR.3.1 Validate: Ensure that the digital signature is correct.

PR.3.2 Remove: Remove storages and streams from documents that are not validated by the digital signature process.

PR.3.3 Replace: Remove the digital signature from the document.

PR.3.4 External Filtering Required: Pass the file to a third party application to validate the signature.

PR.3.5 Review: N/A

PR.3.6 Reject: Reject any file containing digital signatures.

REFERENCE

See [MS-OFFCRYPTO].pdf, Section 2.3.5.4, RC4 CryptoAPI Encryption Summary Stream.

See [MS-OFFCRYPTO].pdf, Section 2.5.1, CryptoAPI Digital Signature Structures and Streams.

See [MS-OFFCRYPTO].pdf, Section 2.5.3, _xmlsignatures Storage.

See [MS-DOC].pdf, Section 4.1 and [MS-OFFCRYPTO].pdf, Sections 4.3-4.5

See [MS-XLS].pdf, Section 2.2.10, Encryption (Password to Open).

See [MS-PPT], Section 2.4.7 ModifyPasswordAtom.

OFFICE 2003.4.24: END

OFFICE 2003.4.25: Color & Size Obfuscation

DESCRIPTION

Some characters appear visually obscured due to the font color matching the background color. The font color of some document text closely matches the background color of the text. This results in text that is not visible in the authoring application.

Some character sizes are outside a certain normal range. The sizes of some characters in the document are below the value defined by the Size Obfuscated Text Minimum or above the value defined by Size Obfuscated Text Maximum.

The item applies to Word, Excel, and PowerPoint (2003 and higher versions).

CONCERN:

Any free-form text fields provide the possibility of data disclosure threats. One should consider instances of color and size obfuscation in Office 2003 documents a data disclosure threat.

RECOMMENDATION:

AR.1 Validate: N/A

AR.2 Remove: Remove text in which background and foreground colors fall with a prescribed range of each other.

AR.3 Remove: Remove text in which background or foreground opacity can cause obfuscation of the text.

AR.4 Remove: Remove text in which the size falls below or above a prescribed value.

AR.5 Remove: Remove images in which the size falls below or above a prescribed value.

AR.6 Replace: Replace background or foreground colors of text in which background and foreground colors fall with a prescribed range of each other to make the text visible.

AR.7 Replace: Adjust the opacity of foreground or background colors in which background or foreground opacity causes obfuscation of the text, such that the text becomes visible.

AR.8 Replace: Readjust the size of text where size falls below or above a prescribed value to a normal value that makes the text readable.

AR.9 External Filtering Required: Pass obfuscated data to the action that is configured for that data type.

AR.10 Review: Adjust text with color or opacity obfuscation and submit for human review.

AR.11 Review: Adjust text with size obfuscation to improve the size and submit for human review.

AR.12 Review: Adjust images with size obfuscation, accounting for scaling and cropping, and submit for human review.

OFFICE 2003.4.25: END

5. ACRONYMS

The following table provides the denotation for the acronyms that appear in this document.

Table 5-1. Acronyms

Acronym	Denotation
aBNF	augmented Backus-Naur Form
ADO	ActiveX Data Object
ANSI	American National Standards Institute
API	Application Programming Interface
AR	All Recommendation
BIFF	Binary Interchange File Format
BLIP	Binary Large Image or Picture
BMP	Bitmap
BNF	Baukus-Naur Form
BOF	Beginning of File
CDS	Cross Domain Solution
CFB	Compound File Binary
CP	Character Positions
DAO	Data Access Object
DB	Database
DIB	Device-Independent Bitmap
DIFAT	FAT Directory Structure
DOS	Disk Operating System
DTG	Data Transfer Guidance
EMF	Enhanced Metafile
ER	Excel Recommendation
FAT	File Allocation Table
FIB	File Information Block
GIF	Graphics Interchange Format
GUI	Graphical User Interface
GUID	Globally unique identifier
IAD	Information Assurance Directorate

Acronym	Denotation
IC	Intelligence Community
ID	Identification
ISG	Inspection and Sanitization Guidance
JPEG	Joint Photographic Experts Group Image
LTR	Left to Right
MAC®	Macintosh® [3]
MAC	Media Access Control
MS	Microsoft
MS-OLDS	Microsoft Object Linking and Embedding Data Structures
MS-PPT	Microsoft PowerPoint
MS-XCL	Microsoft Excel
NSA	National Security Agency
OCR	Optical Character Recognition
ODBC	Open Database Connectivity
ODSO	Office Data Source Object
OLAP	Online Analytical Processing
OLE	Object Linking and Embedding
OLEDS	Object Linking and Embedding Data Structures
PNG	Portable Network Graphics
PR	PowerPoint Recommendation
RC4	Rivest Cipher 4
RTL	Right to Left
SNAC	Systems and Network Analysis Center
Sprm	Single Property Modifiers
SQL	Structured Query Language
SSL	Secure Sockets Layer
SVG	Scalable Vector Graphics
TIFF	Tagged Image File Format
UI	User Interface
URL	Uniform Resource Locator

[3] Macintosh, MAC are registered trademarks of Apple, Inc.

Acronym	Denotation
UTF	Unicode Transformation Format
UUID	Uniquely Universal Identifier
VBA	Visual Basic for Applications
WMF	Windows Metafile
WR	Word Recommendation
XML	Extensible Markup Language
XOR	Exclusive Or

6. REFERENCED DOCUMENTS

The following publications were referenced or used to prepare this document.

RFC- 2234 Augmented BNF.

RFC-2781 Unicode UTC-16.

Red black trees : http://en.wikipedia.org/wiki/Red_black_trees.

"Guidelines for Microsoft Office OpenXML File Analysis and Sanitization Tools," Enterprise Applications Division of the Systems and Network Analysis Center (SNAC), IAD, NSA.

[MS-CFB]: Compound File Binary File Format, Section 2.2 Compound File Header. https://msdn.microsoft.com/enus/library/dd941946%28PROT.10%29.aspx.

[MS-CFB]: Compound File Binary File Format, Section 2.6 Compound File Directory Sectors. https://msdn.microsoft.com/en-us/library/dd942368%28PROT.10%29.aspx.

[RFC-4122], <https://www.ietf.org/rfc/rfc4122.txt >

[MS-SECO]: Windows Security Overview. Section 2.5.5 Globally Unique Identifiers (GUIDs), https://msdn.microsoft.com/en-us/library/cc246025%28PROT.10%29.aspx.

[MS-OLEDS]: Object Linking and Embedding (OLE) Data Structures, Section 2.2.1 PresentationObjectHeader,

https://msdn.microsoft.com/en-us/library/dd942157%28PROT.10%29.aspx

[MS-OLEDS]: Object Linking and Embedding (OLE) Data Structures, Section, 2.2.2 StandardPresentationObject,

https://msdn.microsoft.com/en-us/library/dd942063%28PROT.10%29.aspx

[MS-OLEDS]: Object Linking and Embedding (OLE) Data Structures, Section 2.2.2.1 MetaFilePresentationObject,

https://msdn.microsoft.com/en-us/library/dd941944%28PROT.10%29.aspx

[MS-OLEDS]: Object Linking and Embedding (OLE) Data Structures, Section 2.2.2.2 BitmapPresentationObject,

https://msdn.microsoft.com/en-us/library/dd942474%28PROT.10%29.aspx

[MS-OLEDS]: Object Linking and Embedding (OLE) Data Structures, Section 2.2.2.3 DIBPresentationObject,

https://msdn.microsoft.com/en-us/library/dd942462%28PROT.10%29.aspx

[MS-OLEDS]: Object Linking and Embedding (OLE) Data Structures, Section 2.2.3.3 RegisteredClipboardFormatPresentationObject,

https://msdn.microsoft.com/en-us/library/dd942081%28PROT.10%29.aspx

[MS-OLEDS]: Object Linking and Embedding (OLE) Data Structures, Section 2.2.4 ObjectHeader,

https://msdn.microsoft.com/en-us/library/dd942076%28PROT.10%29.aspx

[MS-OLEDS]: Object Linking and Embedding (OLE) Data Structures, Section 2.3.1 ClipboardFormatOrAnsiString,

https://msdn.microsoft.com/en-us/library/dd942254%28PROT.10%29.aspx

[MS-OLEDS]: Object Linking and Embedding (OLE) Data Structures, Sections 2.3.1 ClipboardFormatOrAnsiString and 2.3.2 ClipboardFormatOrUnicodeString,

https://msdn.microsoft.com/en-us/library/dd942254%28PROT.10%29.aspx

https://msdn.microsoft.com/en-us/library/dd942140%28PROT.10%29.aspx

[MS-OLEDS]: Object Linking and Embedding (OLE) Data Structures, Sections 2.3.3 OLEStream and 2.3.3.1 MONIKERSTREAM,

https://msdn.microsoft.com/en-us/library/dd942499%28PROT.10%29.aspx

https://msdn.microsoft.com/en-us/library/dd942007%28PROT.10%29.aspx

[MS-OLEDS]: Object Linking and Embedding (OLE) Data Structures, Section 2.3.4 OLEPresentationStream,

shttps://msdn.microsoft.com/en-us/library/dd942239%28PROT.10%29.aspx

[MS-OLEDS]: Object Linking and Embedding (OLE) Data Structures, Section 2.3.6 OLENativeStream,

http://msdn.microsoft.com/en-us/library/dd942447%28PROT.10%29.aspx

[MS-OLEDS]: Object Linking and Embedding (OLE) Data Structures, Section 2.3.7 CompObjHeader,

https://msdn.microsoft.com/en-us/library/dd942169%28PROT.10%29.aspx

[MS-OLEDS]: Object Linking and Embedding (OLE) Data Structures, Section 2.3.8 CompObjStream,

https://msdn.microsoft.com/en-us/library/dd941977%28PROT.10%29.aspx

[MS-DOC]: Word Binary File Format (.doc) Structure Specification, Section 2.5 Fib,

https://msdn.microsoft.com/en-us/library/dd949344.aspx

[MS-XLS]: Excel Binary File Format (.xls) Structure Specification, Section 2.4 Records,

https://msdn.microsoft.com/en-us/library/dd947465.aspx

[MS-XLS]: Excel Binary File Format (.xls) Structure Specification, Section 2.4.21 BOF,

https://msdn.microsoft.com/en-us/library/dd906793.aspx

[MS-XLS]: Excel Binary File Format (.xls) Structure Specification, Section 2.1.6 Future Record, https://msdn.microsoft.com/en-us/library/dd773042.aspx.

[MS-PPT]: PowerPoint Binary File Format (.ppt) Structure Specification, Section 2.13.24 RecordType, http://msdn.microsoft.com/en-us/library/dd945336.aspx.

[MS-DOC]: Word Binary File Format (.doc) Structure Specification, Section 2.3.4 Comments,

[MS-DOC]: Word Binary File Format (.doc) Structure Specification, Section 2.8.7 & 2.8.8,

https://msdn.microsoft.com/en-us/library/dd906687.aspx

https://msdn.microsoft.com/en-us/library/dd949442.aspx

https://msdn.microsoft.com/en-us/library/dd950924.aspx

[MS-XLS]: Excel Binary File Format (.xls) Structure Specification, Section 2.5.185 NoteRR, & 2.5.186 NoteSh,

https://msdn.microsoft.com/en-us/library/dd923490.aspx

https://msdn.microsoft.com/en-us/library/dd945371.aspx

[MS-PPT]: PowerPoint Binary File Format (.ppt) Structure Specification, Sections 2.5.25 Comment10Container, 2.5.26 Comment10AuthorAtom, 2.5.27 Comment10TextAtom, 2.5.28 Comment10AuthorInitialAtom, and 2.5.29 Comment10Atom,

https://msdn.microsoft.com/en-us/library/dd921059.aspx

https://msdn.microsoft.com/en-us/library/dd910596.aspx

https://msdn.microsoft.com/en-us/library/dd923619.aspx

https://msdn.microsoft.com/en-us/library/dd910521.aspx

https://msdn.microsoft.com/en-us/library/dd908517.aspx

[MS-DOC]: Word Binary File Format (.doc) Structure Specification, Section 2.2.4 STTB & 2.9.292 SttbSavedBy,

https://msdn.microsoft.com/en-us/library/dd906744.aspx

https://msdn.microsoft.com/en-us/library/dd951971.aspx

[MS-DOC]: Word Binary File Format (.doc) Structure Specification, Sections 2.9.273 SttbfAssoc, https://msdn.microsoft.com/en-us/library/dd911026.aspx.

[MS-XLS]: Excel Binary File Format (.xls) Structure Specification, Sections 2.4.244 SCENARIO & 2.4.246 ScenMan,

https://msdn.microsoft.com/en-us/library/dd950118.aspx

https://msdn.microsoft.com/en-us/library/dd926420.aspx

[MS-OSHARED]: Office Common Data Types and Objects Structure Specification,

https://msdn.microsoft.com/en-us/library/cc313156.aspx

[MS-DOC]: Word Binary File Format (.doc) Structure Specification, Sections 2.3.4 Comments, 2.8.7 PlcfandRef, & 2.8.8 PlcfandTxt,

https://msdn.microsoft.com/en-us/library/dd906687.aspx

https://msdn.microsoft.com/en-us/library/dd949442.aspx

https://msdn.microsoft.com/en-us/library/dd950924.aspx

[MS-XLS]: Excel Binary File Format (.xls) Structure Specification, Sections 2.4.136 Header, 2.4.137 HeaderFooter, 2.5.134 FrtFlags, & 2.5.135 FrtHeader,

https://msdn.microsoft.com/en-us/library/dd773041.aspx

https://msdn.microsoft.com/en-us/library/dd907485.aspx

https://msdn.microsoft.com/en-us/library/dd907485.aspx

https://msdn.microsoft.com/en-us/library/dd910234.aspx

[MS-PPT]: PowerPoint Binary File Format (.ppt) Structure Specification, Section 2.4.15 Bar, https://msdn.microsoft.com/en-us/library/dd907451.aspx.

[MS-DOC]: Word Binary File Format (.doc) Structure Specification, Sections 1.3.3 Formatting & 2.6.1 Character Properties,

https://msdn.microsoft.com/en-us/library/dd953675.aspx

https://msdn.microsoft.com/en-us/library/dd947480.aspx

[MS-XLS]: Excel Binary File Format (.xls) Structure Specification,

https://msdn.microsoft.com/en-us/library/cc313154.aspx

[MS-PPT]: PowerPoint Binary File Format (.ppt) Structure Specification, Section 2.6.6 SlideShowSlideInfoAtom, Section 2.8.2 AnimationInfoAtom, Section 2.4.12 PrintOptionsAtom,

https://msdn.microsoft.com/en-us/library/dd943408.aspx

https://msdn.microsoft.com/en-us/library/dd945943.aspx

https://msdn.microsoft.com/en-us/library/dd922802.aspx

[MS-DOC]: Word Binary File Format (.doc) Structure Specification, Sections 2.2.4 STTB, 2.9.227 RmdThreading, 2.9.273 SttbfAssoc, & 2.9.287 SttbfRMark,

https://msdn.microsoft.com/en-us/library/dd906744.aspx

https://msdn.microsoft.com/en-us/library/dd773215.aspx

https://msdn.microsoft.com/en-us/library/dd911026.aspx

https://msdn.microsoft.com/en-us/library/dd953637.aspx

[MS-OLEPS]: Object Linking and Embedding (OLE) Property Set Data Structures, Sections 3.1.4 PIDSI_AUTHOR & 3.1.8 PIDSI_LASTAUTHOR,

https://msdn.microsoft.com/en-us/library/dd942296%28PROT.10%29.aspx

https://msdn.microsoft.com/en-us/library/dd942134%28PROT.10%29.aspx

[MS-XLS]: Excel Binary File Format (.xls) Structure Specification, Sections 2.4.244 SCENARIO, 2.4.340 UsrInfo, 2.5.239 ShortDTR, & 2.5.241 SLCO8,

https://msdn.microsoft.com/en-us/library/dd950118.aspx

https://msdn.microsoft.com/en-us/library/dd920434.aspx

https://msdn.microsoft.com/en-us/library/dd910093.aspx

https://msdn.microsoft.com/en-us/library/dd910232.aspx

[MS-PPT]: PowerPoint Binary File Format (.ppt) Structure Specification, Sections 2.3.2 CurrentUserAtom & 2.4.17.17 BCUserNameAtom,

https://msdn.microsoft.com/en-us/library/dd948895.aspx

https://msdn.microsoft.com/en-us/library/dd945270.aspx

[MS-DOC]: Word Binary File Format (.doc) Structure Specification, Section 2.9.230 RouteSlipInfo, https://msdn.microsoft.com/en-us/library/dd925045.aspx.

[MS-XLS]: Excel Binary File Format (.xls) Structure Specification, Sections 2.4.91DocRoute & 2.4.216 RecipName,

https://msdn.microsoft.com/en-us/library/dd950092.aspx

https://msdn.microsoft.com/en-us/library/dd921365.aspx

[MS-PPT]: PowerPoint Binary File Format (.ppt) Structure Specification, Sections 2.11.1 DocRoutingSlipAtom, & 2.11.2 DocRoutingSlipString,

https://msdn.microsoft.com/en-us/library/dd908875.aspx

https://msdn.microsoft.com/en-us/library/dd922752.aspx

[MS-DOC]: Word Binary File Format (.doc) Structure Specification, Sections 2.9.208 PrDrvr, 2.9.209 PrEnvLand, & 2.9.210 PrEnvPort,

https://msdn.microsoft.com/en-us/library/dd926965.aspx

https://msdn.microsoft.com/en-us/library/dd905563.aspx

https://msdn.microsoft.com/en-us/library/dd909115.aspx

[MS-XLS]: Excel Binary File Format (.xls) Structure Specification, Section 2.4.199 Pls,

https://msdn.microsoft.com/en-us/library/dd909381.aspx

https://msdn.microsoft.com/en-us/library/dd183565%28VS.85%29.aspx

[MS-OSHARED]: Office Common Data Types and Objects Structure Specification, Section 2.3.4 SmartTag Objects,

https://msdn.microsoft.com/en-us/library/dd946805.aspx

[MS-DOC]: Word Binary File Format (.doc) Structure Specification, Section 2.9.248 SmartTagData,

https://msdn.microsoft.com/en-us/library/dd911057.aspx

[MS-XLS]: Excel Binary File Format (.xls) Structure Specification, Sections 2.5.110 FactoidData, 2.5.125 FeatSmartTag, 2.5.134 FrtFlags, 2.5.135 FrtHeader, & 2.4.111 Feat,

https://msdn.microsoft.com/en-us/library/dd945735.aspx

https://msdn.microsoft.com/en-us/library/dd908824.aspx

https://msdn.microsoft.com/en-us/library/dd946080.aspx

https://msdn.microsoft.com/en-us/library/dd910234.aspx

https://msdn.microsoft.com/en-us/library/dd911261.aspx

[MS-PPT]: PowerPoint Binary File Format (.ppt) Structure Specification, Sections 2.4.23.7 PP11DocBinaryTagExtension, & 2.11.28 SmartTagStore11Container,

https://msdn.microsoft.com/en-us/library/dd907573.aspx

https://msdn.microsoft.com/en-us/library/dd911007.aspx

[MS-DOC]: Word Binary File Format (.doc) Structure Specification, Section 2.2.6, Encryption and Obfuscation (Password to Open),

https://msdn.microsoft.com/en-us/library/dd922770.aspx

[MS-XLS]: Excel Binary File Format (.xls) Structure Specification, Section 2.2.10, Encryption (Password to Open),

https://msdn.microsoft.com/en-us/library/dd905723.aspx

[MS-OFFCRYPTO]: Office Document Cryptography Structure Specification, Section 2.3.5 Office Binary Document RC4 CryptoAPI Encryption & Section 2.5.1 CryptoAPI Digital Signature Structures and Streams,

https://msdn.microsoft.com/en-us/library/dd905225.aspx

https://msdn.microsoft.com/en-us/library/dd953667.aspx

[MS-PPT]: PowerPoint Binary File Format (.ppt) Structure Specification, Section 2.4.7 ModifyPasswordAtom, https://msdn.microsoft.com/en-us/library/dd908152.aspx

OpenOffice.org's Documentation of the Microsoft Compound Document File Format,

https://sc.openoffice.org/compdocfileformat.pdf

7. VALIDATING CFB FILES

This section provides an overview for validating CFB files. It describes CFB sector sizes, integrity checking rules, FAT verification, directory verification, and file residual data checks.

7.1 A Note on Sector Sizes

Though the CFB specification specifies two sector sizes, 4096 bytes (CFB header sector shift o 0x000C) and 512 bytes,(sector shift of 0x0009), a sector size of 4096 bytes is rarely, *if ever* seen by file inspection and sanitization programs. Anecdotal information from vendors, including Microsoft, indicates that a sector size or 4096 has never been seen. However, because a sector size of 4096 is part of the standard, programs should have the ability to process and handle such files.

7.2 Integrity Checking Rules

This section discusses integrity checking rules in terms of the following:

- File size checks
- Header integrity checks
- DIFAT loading and checking
- FAT loading and checking
- Mini FAT loading and checking
- Directory loading and checking
- Mini stream loading and checking

7.2.1 File Size Checks

A CFB document must have a file size of *nn* integer number of 512 byte sectors. However, because a sector size of 4096 bytes is rare, it is unclear as to if the sector containing the CFB header is 4096 bytes or the true size of the CFB header, which is 512 bytes. Failure of this rule indicates a corrupt, rejected file. *[MS-CFB] 2.2.*

7.2.2 Header Integrity Checks

The Compound File Header should be checked for good and consistent values in the following manner:

1. *Compound File Header.Header Signature* must match the documented signature of 0xD0, 0xCF, 0x11, 0xE0, 0xA1, 0xB1, 0x1A, 0xE1. Failure of this rule indicates either the file is corrupt or not a CFB. *[MS-CFB] 2.2.*

2. *Compound File Header.Header CLSID* must be set to all zeros. Failure of this rule indicates either the file is corrupt or not a CFB. *[MS-CFB] 2.2.*

3. *Compound File Header.Minor Version* must be 0x003E. Failure of this rule indicates either the file is corrupt or not a CFB. *[MS-CFB] 2.2.*

4. *Compound File Header.Major Version* must be 0x0003 (version 3) or 0x0004 (version 4). Failure of this rule indicates either the file is corrupt or not a CFB. *[MS-CFB] 2.2.*

5. *Compound File Header.Byte Order* must be 0xFFFE. Failure of this rule indicates either the file is corrupt or not a CFB. *[MS-CFB] 2.2.*

6. *Compound File Header.Sector Shift* must be 0x0009 (if Major Version is 0x0003) or 0x000C (if Major Version is 0x0004). Failure of this rule indicates either the file is corrupt or not a CFB. *[MS-CFB] 2.2.*

7. *Compound File Header.Mini Sector Shift* must be 0x0006 (indicating 64 bytes mini sectors). Failure of this rule indicates either the file is corrupt or not a CFB. *[MS-CFB] 2.2.*

8. *Compound File Header.Reserved* must be all zeros. Failure of this rule indicates either the file is corrupt or not a CFB. *[MS-CFB] 2.2.*

9. *Compound File Header.Number of Directory Sctors* must be zero in major version 3 files. In version 4 files, it is the number of Directory Sectors. Failure of this rule indicates either the file is corrupt or not a CFB. Note that version 4 files have rarely or never been seen so it is unclear what consistency checking can be performed on this value if a version 4 file is encountered. *[MS-CFB] 2.2. No*

10. *Compound File Header.Number of FAT Sectors* contains the number of sectors in the compound file. This number should not be used to preallocate space for the FAT. Consistency between the real size of the FAT and this number can only be

performed after loading the FAT. It also should be no larger than the number of sectors comprising the file. *[MS-CFB] 2.2.*

11. C*ompound File Header.First Directory Sector Location* should be in the range of 0 to the number of sectors constituting the file minus 1. Failure of this rule indicates either the file is corrupt or not a CFB, and the file is rejected. *[MS-CFB] 2.2.*

12. *Compound File Header.Transaction Signature Number*. It is unclear if any consistency checking can be performed on this field. It should be cleared to 0 in sanitization cases. *[MS-CFB] 2.2.*

13. *Compound File Header.Mini Stream Cutoff Size* must be set to 0x00001000. Failure of this rule indicates either the file is corrupt or not a CFB. *[MS-CFB] 2.2.*

14. *Compound File Header.First Mini FAT Sector Location* must fall in the range of zero to the number of sectors comprising the file minus one. Failure of this rule indicates either the file is corrupt or not a CFB. It also should be no larger than the number of sectors constituting the file or the value should be ENDOFCHAIN. In this case the *Compound File Header Number of Mini FAT Sectors* value should be zero. *[MS-CFB] 2.2.*

15. *Compound File Header.Number of Mini FAT Sectors* contains the number of sectors that comprise the Mini FAT. The value should not be used to preallocate space for the Mini FAT. Consistency between the real size of the Mini FAT and this value can only be determined when loading the FAT and then loading the Mini FAT. It also should be no larger than the number of sectors comprising the file. *[MS-CFB] 2.2.*

16. *Compound File Header.First Mini DIFAT Sectors* must be in the range of zero to the number of sectors constituting the file minus one. Failure of this rule indicates either the file is corrupt or not a CFB, and the file is rejected or the value should be ENDOFCHAIN. In this case the *Compound File Header.Number of DIFAT Sectors* value should be zero. *[MS-CFB] 2.2 and [MS-CFB] 2.1.*

17. *Compound File Header.Number of DIFAT Sectors* contains the number of sectors that constitute the DIFAT. The value should not be used to preallocate space for the DIFAT. Consistency between the real size of the DIFAT and the value can only be determined when loading the DIFAT. It also should be no larger than the number of sectors constituting the file. *[MS-CFB] 2.2.*

18. *Compound File Header.DIFAT* contains the first 109 entries in the DIFAT. Each entry in this array should range from 0 to the number of sectors constituting the

file minus one. Each entry in the Header DIFAT should range from zero to the number of sectors constituting the file or FREESEC. Failure of this rule indicates either the file is corrupt or not a CFB. In cases where *Compound File Header.Number of DIFAT Sectors* is less than or equal to 109, only the first *Compound File Header.Number of DIFAT Sectors* are sector indexes. The rest must be set to SECTFREE. [*MS-CFB] 2.2, and [MS-CFB] 2.1*

It is unclear when *Compound File Header.Major Version* == 0x0004 with *Compound File Header.Sector Shift* == 0x000C if the sector size containing the header is 512 bytes or 4096 bytes. If the sector size is 512 bytes then the header fills the sector. If the sector size is 4096 bytes, then there are 3584 bytes that can be used to hide data in the header.

7.2.3 DIFAT Loading and Checking

[MS-CFB] 2.5 details the DIFAT structure. Loading the DIFAT is a three stage operation. The first 109 entries of the DIFAT are located in *Compound File Header.DIFAT.* Any other needed entries reside in a chain of sectors that starts at *Compound File Header.First Mini DIFAT Sector* for *Compound File Header.Number of DIFAT Sectors*. Each Mini DIFAT Sector is a list of 127 (version 3) or 1023 (version 4) FAT sector locations with a link to the next Mini DIFAT Sector. The last Mini DIFAT sector will have *DIFAT.Next DIFAT Sector Location* set to ENDOFCHAIN. A suggestion is to compute the length of the DIFAT chain. It must equal *Compound File Header.Number of DIFAT sectors*. Then one can allocate space for the DIFAT, and load and verify the DIFAT as follows:

1. If the computed chain length of the Mini DIFAT chain is greater than the number of sectors constituting the file, then reject the file.

2. If the computed chain length of the Mini DIFAT chain is not equal to *Compound File Header.Number of DIFAT Sectors*, then reject the file.

3. Each *DIFAT.FAT Sector Location* must range between zero to the number of sectors constituting the file or be set to FREESECT. Failure of this rule indicates either the file is corrupt or not a CFB.

4. *DIFAT.Next DIFAT Sector* must range between zero to the number of sectors constituting the file, or if this is the last DIFAT sector in the chain, set to ENDOFCHAIN. Failure of this rule indicates either the file is corrupt or not a CFB.

   [MS-CFB] 2.5.

After loading the DIFAT, one can perform several consistency checks:

1. If the number of DIFAT entries does not cover the number of sectors constituting the file, the file is corrupt or not a CFB.

2. If the number of entries per sector times *Compound File Header.Number of FAT Sectors* does not cover the number of sectors constituting the file, then file is corrupt or not a CFB file.

3. If *Compound File Header.Number of FAT Sectors* is greater than the number of DIFAT entries, the file is corrupt or not a CFB file.

4. The first *Compound File Header.Number of FAT Sectors* entries in the DIFAT must range between 0 to the number of sectors constituting the file. The rest must be set to FREESECT. Failure of this rule indicates either the file is corrupt or not a CFB.

After the DIFAT is loaded the FAT, mini FAT, and the Directory can be loaded.

7.2.4 FAT Loading and Checking

The FAT is an array of sector indexes for managing the space represented by the file. The FAT is used to chain streams and to indicate used and free sectors. After loading and verifying the DIFAT, one can load and *partially* verify the FAT. Each FAT sector contains either 128 FAT entries (version 3 files) or 1024 FAT entries (version 4 files). After loading the FAT, the following verification should occur:

1. Each FAT entry must range between zero to the number of sectors constituting the file or be one of DIFSEC, FATSEC, ENDOFCHAIN, or FREESECT. Failure of this rule indicates either the file is corrupt or not a CFB.

Additional validation needs to be performed on the FAT; however, this cannot occur until after loading other structures from the file. Also, one needs the FAT to load them, and must take care when using the FAT to load the mini-FAT and the directory.

[MS-CFB] 2.3

7.2.5 Mini FAT Loading and Checking

The Mini FAT is an area for saving space in streams that would be wasted when using regular sectors. The size of Mini FAT sectors is 64 bytes, as compared to 512 bytes (version 3) or 4096 bytes (version 4). The *Compound File Header.Mini Stream Cutoff Size* is used to determine if a stream consists of sectors or mini sectors. The structure of the

Mini FAT is similar to the FAT, except the Mini FAT entries are indexes in the mini stream.

1. When loading the mini FAT, the number of sectors loaded must equal *Compound File Header.Number of Mini FAT Sectors*. Failure of this rule indicates either the file is corrupt or not a CFB.

2. Do not use *Compound File Header.Number of Mini FAT Sectors* to preallocate space for the Mini FAT. At this point, one has not validated the FAT and a corrupted or maliciously created file could cause problems.

3. After loading, each entry in the Mini FAT must range between zero to the number of Mini FAT entries (the actual size of the mini stream is unknown at this point), ENDOFCHAIN or FREESECT. Failure of this rule indicates either the file is corrupt or not a CFB.

7.2.6 Directory Loading and Checking

The Directory provides a file system type structure to the streams that compose the compound file. Using *Compound File Header.First Directory Sector Location* and *Compound File Header.Number of Directory Sectors* load the Directory. Because the FAT remains unvalidated here, the number of sectors constituting the directory must not exceed *Compound File Header.Number of Directory Sectors* (in version 4 files). In version 3 files the number of sectors constituting the directory must not exceed the number of sectors constituting the file. Also, do not use *Compound File Header.Number of Directory Sectors* to preallocate directory space. After loading the directory, one can traverse the directory according to the red-black tree algorithm to find the Root Entry.

7.2.7 Mini Stream Loading and Checking

The stream that is associated with the Root Entry is the Mini Stream. Locate the Root Entry and load the Mini Stream using *Root Entry.Starting Sector Location*. Because FAST validation has not occurred yet, the number of sectors constituting the Mini Stream must not exceed the number of sectors constituting the file. Failure of this rule indicates either the file is corrupt or not a CFB.

The size of the Mini Stream must remain consistent with *Root Entry.Stream Size*. Failure of this rule indicates either the file is corrupt or not a CFB. It is possible that the file contains no Mini Stream.

[MS-CFB] 2.6.2

7.3 FAT Verification

After loading all CFB components, as detailed in Section 7.2, one can perform final verification of the FAT, noting these rules:

1. Each DIFAT sector must be marked DIFSECT in the FAT. Failure of this rule indicates either the file is corrupt or not a CFB.

2. Each FAT sector must be marked FATSECT in the FAT. Failure of this rule indicates either the file is corrupt or not a CFB.

3. Each FAT entry must be referenced once and only once or be marked FREESECT. Failure of this rule indicates either the file is corrupt or not a CFB.

To verify the FAT:

1. Create an array of integers sized to the number of FAT entries and initialized to zero, and called the reference array.

2. For the *Compound File Header.First Mini FAT Sector Location*, increment the corresponding entry in the reference array.

3. For the *Compound File Header.First Directory Sector Location*, increment the corresponding entry in the reference array.

4. For each DIFAT sector, increment the corresponding entry in the reference array.

5. For each FAT sector, increment the corresponding entry in the reference array.

6. For each FAT entry, if the value is not FREESECT, ENDOFCHAIN, DIFSECT, or FATSECT, increment the corresponding entry in the reference array.

7. For each Directory entry, if it is the Root Entry or a Stream Entry and the *Compound File Directory Entry.Stream Size* is greater than or equal to *Compound File Header.Mini Stream Cutoff Size*, using the *Compound File Directory Entry.Starting Sector Location* increment the corresponding entry in the reference array.

8. Now, do the check using the reference array. Each entry in the reference array must be either zero or one. And, each zero must be marked FREESECT in the FAT. Also, for each entry in the FAT, if the value is FREESECT, the corresponding entry in the reference array must be marked zero.

This gives confidence that the FAT has no chain loops and no unaccounted for or unreferenced sectors.

One must perform a similar check on the Mini FAT:

1. Create a reference array of integers, initialized to zero, and sized to the number of entries in the mini-FAT.

2. For each Mini FAT entry that does not have a value of FREESEC or ENDOFCHAIN, increment the corresponding entry in the reference array.

3. For each Directory entry that is a Stream Entry and the *Compound File Directory Entry.Stream Size* is less than *Compound File Header.Mini Stream Cutoff Size*, using the *Compound File Directory Entry.Starting Sector Location* increment the corresponding entry in the reference array.

4. Now, conduct the check using the reference array. Each entry in the reference array must either be zero or one, and each zero must be marked FREESECT in the Mini FAT. Also, for each Mini FAT entry with a value is FREESECT, the corresponding entry in the reference array must be marked zero.

This gives confidence that the Mini FAT has no chain loops and no unaccounted for or unreferenced sectors.

7.4 Directory Verification

The directory is a tree of streams and storages (files and directories, respectively) that is organized in a red-black balanced tree. Thus, each used directory entry must be referenced only once. Unused directory entries must not be referenced. Programs should validate that the streams sizes are consistent with the sector or mini sector chains that constitute the stream, noting these rules:

1. Only one Directory entry should be marked as the Root Entry. Failure of this rule indicates either the file is corrupt or not a CFB.

2. Validate that each used Directory entry is only referenced one time. Validate that each unused Directory entry is not referenced. Failure of this rule indicates either the file is corrupt or not a CFB.

   One can perform this validation using a reference array and incrementing each entry for each *Directory Entry.Left Sibling* not equal to NOSTREAM, and for each *Directory Entry.Right Sibling*. Then, check each entry in the reference array for zero or one.

Root Entry directory entries must have zero references. Unused directory entries must have zero references. All others must have one reference. This gives some level of confidence that the directory tree is properly constructed.

1. Each unused Directory entry should not contain residual data.

2. Note that older versions of the Office software (prior to Office 2003) were more remiss about leaving residual data in unused stream entries. In a sanitation case, one should clear unused Directory entries.

3. For each Directory entry that is a Root Entry or Stream Entry, compute the actual stream size using the *Compound File Directory Entry.Starting Sector Location* and chaining through the sectors or mini sectors constituting the stream. The *Compound File Directory Entry.Stream Size* must be less or equal to the actual stream size. Failure of this rule indicates either the file is corrupt or not a CFB.

7.5 File Residual Data Check

The sectors marked *unused* in the FAT and the mini sectors marked *unused* in the Mini FAT can contain residual data. MS Office 2003 creates Word documents without residual data. However, previous versions of Word, Excel 2003 (and prior versions) and PowerPoint 2003 (and prior versions) were less rigorous about leaving residual data in documents. Therefore, a strict check to ensure unused sectors are zero will generate false positives and fail many documents (this may be desirable in sanitizing files).

In Sector residual data, two different cases apply. The first, termed internal residual, occurs when used sectors bracket unused sectors. The second, terminal residual, involves unused sectors after the last used sector.

MS Office document clients can leave internal residual data in documents. Usually, this does not indicate an attempt to maliciously hide data. However, one can hide data maliciously in internal unused sectors. One should clear unused sectors always.

MS Office document clients rarely, if ever, leave terminal residual data. The FAT does usually describe more space than the file actually occupies. Therefore, it becomes easy to append data (up to a certain size) to an Office document and have it carried along. File inspection and sanitization programs can either enforce a rule prohibiting any terminal unused sectors or setting all unused terminal sectors to zero.

8. LIST OF FIELD CODES

The following table lists the field codes used in this document.

Table 8-1 List of Field Codes

Field	Field's Content or Action It Takes
AddressBlock	Insert a mail merge address block
Advance	Offset subsequent text within a line to the left, right, up, or down
Ask	Prompt the user for text to assign to a bookmark
Author	The name of the document's author from Summary Info
AutoNum	Insert an automatic number
AutoNumLgl	Insert an automatic number in legal format
AutoNumOut	Insert an automatic number in outline format
AutoText	Insert an Auto Text entry
AutoTextList	Insert auto text based on style
BarCode	Insert a delivery point bar code
BidiOutline	Sets Outline to Right to Left for bi-directional languages such as Hebrew and Arabic
Comments	The comments from Summary Info
Compare	Compare two values and return the numeric value 1 if the comparison is true or 0 (zero) if the comparison is false
CreateDate	The date the document was created
Database	Insert data from an external database
Date	Today's date
DocProperty	Insert the value of the File Property chosen in Options
DocVariable	Insert the value of the document variable named NAME
EditTime	The total document editing time
Eq	Create a scientific equation
FileName	The documents name and location
FileSize	The size on disk of the active document
Fill-in	Prompt the user for text to insert in the document
Formula	Calculate the result of an expression
GoToButton	Move the insertion point to a new location
GreetingLine	Insert a mail merge field
Hyperlink	Open and jump to the specified file

Field	Field's Content or Action It Takes
If	Evaluate arguments conditionally
IncludePicture	Insert a picture from a file
IncludeText	Insert text from a file
Index	Create an index
Info	Data from Summary Info
Keywords	The keywords from Summary Info
LastSavedBy	Name of the user who last saved the document
Link	Insert part of a file by using OLE
ListNum	Insert an element in a list
MacroButton	Run a macro
MergeField	Insert a mail merge field
MergeRec	The number of the current merge record
MergeSeq	Merge record sequence number
Next	Go to the next record in a mail merge
NextIf	Conditionally go to next record in a mail merge
NoteRef	Insert the number of a footnote or endnote
NumChars	The number of characters in the document
NumPages	The number of pages in the document
NumWords	The number of words in the document
Page	Insert the number of the current page
PageRef	Insert the number of the page containing the specified bookmark
Print	Download commands to a printer
PrintDate	The date the document was last printed
Private	Store data for documents converted from other file formats
Quote	Insert literal text
RD	Create an index, table of contents, table of figures, and/or table of authorities by using multiple documents
Ref	Insert the text marked by a bookmark
RevNum	Insert the number of times the document has been saved
SaveDate	The date the document was last saved
Section	Insert the number of the current section
SectionPages	Insert the total number of pages in the section

Field	Field's Content or Action It Takes
Seq	Insert an automatic sequence number
Set	Assign new text to a bookmark
SkipIf	Conditionally skip a record in a mail merge
StyleRef	Insert the text from a like-styled paragraph
Subject	The document's subject from Summary Info
Symbol	Insert a special character
TA	Mark a table of authorities entry
TC	Mark a table of contents entry
Template	The name of the template attached to the document
Time	The current time
Title	The document's title from the Summary Info
TOA	Create a Table of Authorities
TOC	Create a Table of Contents
UserAddress	Address from Tools Options User Info
UserInitials	Initials from Tools Options User Info
UserName	Name from Tools Options User Info
XE	Mark an index entry

Appendix A

BNF Constructs

APPENDIX A: BNF CONSTRUCTS

BNF Constructs

Explanation of Construct Grammar

This document uses the "augmented Baukus-Naur Form" (aBNF) to describe construct grammar. Because we describe binary data, we extend the grammar definitions of standard aBNF. This extension should give readers a sufficiently accurate portrayal of the binary data. In particular, users should note the following:

- Comments start with '**<!--**'and close with '**-->**', rather than proceeding with a single '**;**'.
- Hexadecimal integer representations use standard C programming language notation of 0x.
- In some instances, the document uses standard C programming language to denote an array of items (e.g., '[5]'). In other instances it uses the aBNF notation (e.g., of '1*5').

Common aBNF Productions

Many of the Office 2003 constructs use similar data structures. This is true especially for low level data, such as strings and integers. This section defines some of the common structures.

```
<Integer64> ::=  <octet>[8]

<Integer32> ::= <octet>[4]

<Integer16> ::= <octet>[2]

<Integer16> ::= 0x0000

<Integer8> ::= <octet>

<octet> ::= <bit>[8]

<octet-zero> ::= 0x00

<octet-ignored> ::= <octet> <!-- must be ignored -->

<octet-zero-ignored> ::= <octet-zero> <!-- must be ignored -->

<octet-boolean> ::= <0x00> | <0x01>

<octet-null-terminated> ::= NULL | <octet> <octet-null-terminated>

<bit> ::= 0 | 1

<bit-zero> ::= 0

<bit-zero-ignored> ::= <bit-zero> <!-- must be ignored -->

<bit-ignored> ::= <bit> <!-- must be ignored -->

<variable-empty> ::= <!-- empty -->

<GUID> ::= <octet>[16]

<Windows FILETIME> ::= <Integer32> <!-- dwLowDateTime -->  <Integer32>
     <!-- dwHighDateTime -->

<LengthPrefixAnsiString> :: = <Length> <AnsiString>

<Length> ::= <Integer32>

<AnsiString> ::= 0*<AnsiCharacter> <octet-zero>

<AnsiCharacter> ::= 0x20 - 0xFF

<LengthPrefixUnicodeString ::= <Integer32> <UnicodeString>

<UnicodeString> ::= 0*<UnicodeCharacters><Integer16-zero>

<UnicodeCharacter> ::= 0x0000-0xFFFF <!-- possible unicod character range
     -->
```

```
<OLE_1_0_Version> :: <Integer32> <!-- must be ignored -->

<ClassName>  ::= <LengthPrefixAnsiString>

<FILETIME> ::= <Integer32>[2]
```

Compound File Binary (CFB) *(See Section 3 in main document.)*

OFFICE 2003.3.1: CFB Header

PRODUCT: WORD, EXCEL, POWERPOINT

GRAMMAR:

```
<Compound File Header>::=<signature> <Header Class ID> <Minor Version>
     <Version Specific Information> <Number of FAT Sectors> <First Dir
     Location>  <Transaction Signature Number> <Minimum Stream Cutoff
     Size> <First Mini FAT Sector Location> <Number of Mini FAT Sectors>
     <First Mini DIFAT Location> <Number of DIFAT Sectors> <DIFAT>

<Signature> ::= 0xD0 0xCF 0x11 0xE0 0xA1 0xB1 0x1A 0xE1

<Header Class ID> ::=  <Integer32> <Integer32> <Integer32> <Integer32> <-
     - GUID, See Reference section below for more information -->

<Minor Version> ::= 0x003E

<Version Specific Information> ::=  <Major Version 3> |  <Major Version
     4>

<Major Version 3> ::=  0x0003 <Byte Order> <Sector Shift 3> <Reserved>
     <Number of Directory Sectors 3>

<Major Version 4> ::= 0x0004 <Byte Order> <Sector Shift 4> <Reserved>
     <Number of Directory Sectors 4>

<Byte Order> ::= 0xFFFE

<Sector Shift 3> ::= 0x0009 <!-- 512 bytes (2^9) -->

<Sector Shift 4> ::= 0x000C <!-- 4096 bytes (2^12) -->

<Reserved> ::= <Integer16> <Integer32>

<Number of Directory Sectors 3> ::= 0x00000000

<Number of Directory Sectors 4> ::= <Integer32>
```

```
<Number of FAT sectors> :: = <Integer32>

<First Dir Sector> ::= <Integer32> <!-- Indicates the sector number for
     the directory stream -->

<Transaction Signal Number ::= <Integer32> <!-- Optional field for
     tracking saves -->

<Minimum Sector Cutoff  Size> ::= 0x00000100 <!-- Represents the cutoff
     size for using FAT, rather than Mini FAT sectors -->

<First Mini FAT Sector Loc> ::= <Integer32> <!-- The offset to the first
     Mini FAT sector>

<Number of Mini FAT Seectors> ::= <Integer32>

<First Mini DIFAT Location> ::= <Integer32> <!-- DIFATs entries store
     locations of FAT sectors>

<Number of DIFAT Sectors> ::= <Integer32>

<DIFAT> ::= <Integer32>[109]
```

OFFICE 2003.3.1 END

OFFICE 2003.3.2: Directory and Directory Entry

PRODUCT: WORD, EXCEL, POWERPOINT

GRAMMAR:

```
<Directory> ::= <Directory Entry> | <Directory Entry>+

<Directory Entry> ::= <Directory Entry Name> <Directory Entry Name
     Lengths> <Object Type> <Color Field> <Left Sibling ID> <Right
     Sibling ID> <Child ID> <ClsID> <State Bits> <Create Time> <Modified
     Time> <Starting Sector Location> <Stream Size>

<Directory Entry Name> ::= <UTF-16 Unicode String>

<UTF-16 Unicode String> ::= <UTF-16>[63] <UTF-16 NULL> <!-- max is 64
     bytes -->

<UTF-16> ::= <Integer16> <!-- Encoding rules of UTF-16 are beyond the
     scope of this definition, but see XXXX for more details -->

<Directory Entry name Length  ::= <Integer16>

<Object Type> ::= 0x00 <!-- Unknown or unallccoated --> | 0x0 <!--
     Storage Object --> | 0x02 <!- Stream Object --> | 0x05 <!-- Root
```

```
                    Storage Object -->

               <Color Flag> ::= 0x00 <!-- red --> | 0x01 <!-- black -->

               <left Side Sibling ID> :: = <Stream ID> | <NOSTREAMID>

               <Right Side Sibling ID> ::=  <Stream ID> | <NOSTREAMID>

               <Child ID> ::= Stream ID> | <NOSTRAMIDID>

               <Stream ID> ::=  0x00000000 – 0xFFFFFFF9

               <NOSTREAMID> ::= 0xFFFFFFFF

               <ClsID> ::= <GUID>

               <State Bits> ::= <Integer32>

               <Creation Time> ::= <Windows FILETIME>

               <Modified Time> ::= <Windows FILETIME>

               <Starting Sector Location> ::= <Integer32>

               <Stream Size> ::= <Integer54>
```

OFFICE 2003.3.2 END

OFFICE 2003.3.3: Microsoft GUID

PRODUCT: WORD, EXCEL, POWERPOINT

GRAMMAR:

```
               <GUID> :: = <Data1> <Data2> <Data3> <Data4>

               <Data1> ::= <Integer32> <!-- also known as “time-low” -->

               <Data2> ::= <Integer16> <!-- time-mid -->

               <Data3> ::= <Integer16> <!-- time-high-and-version -->
```

OFFICE 2003.3.3 END

Embedded and Linked Objects *(See Section 3 in main document.)*

OLE 1.0 Constructs:

OFFICE 2003.3.4: Embedded Objects: Not Applicable (N/A)

OFFICE 2003.3.4 END

OFFICE 2003.3.5: Presentation Object Header

PRODUCT: WORD, EXCEL, POWERPOINT

GRAMMAR:

```
<PresentationObjectHeader> ::= <OLEVersion> <FormatID> <ClassName>

<FormatID> ::= 0x00000000 | 0x00000005 <!-- if 0x00000005 ClassName must
     be present; if 0x00000000, ClassName must NOT be present -->
```

OFFICE 2003.3.5 END

OFFICE 2003.3.6: Standard Presentation Object

PRODUCT: WORD, EXCEL, POWERPOINT

GRAMMAR:

```
<StandardPresentationObject> :: = <Header>  <Width> <Height>

<Header> ::= <PresentationObjectHeader>

<Width> ::= <Integer32>

<Height> ::= <Integer32>
```

OFFICE 2003.3.6 END

OFFICE 2003.3.7: Metafile Presentation Object

PRODUCT: WORD, EXCEL, POWERPOINT

GRAMMAR:

```
<MetaFilePresentationOBject> ::= <Header> <PresentationDataSize>
     <Reserved1> <Reserved2> <Reserved3> <Reserved4> <PresentationData>

<Header> ::= <StandardPresentationObject>

<Reserved1> ::= <Integer16>
```

```
<Reserved2> ::= <Integer16>

<Reserved3> ::= <Integer16>

<Reserved4> ::= <Integer16>

<PresentationData> ::= 1*<octet>
```

OFFICE 2003.3.7 END

OFFICE 2003.3.8: Bitmap Presentation Object

PRODUCT: WORD, EXCEL, POWERPOINT

GRAMMAR:

```
<BitmapPresentationObject> ::= <Header> <PresenationDataSize> <Bitmap>

<Header> ::= <StandardPresentationObject>

<PresentationDataSize> :: Integer32>

<Bimtap> :: = 1*<octet>
```

OFFICE 2003.3.8 END

OFFICE 2003.3.9: DIB Presentation Object

PRODUCT: WORD, EXCEL, POWERPOINT

GRAMMAR:

```
<DIBPresentationObejct> ::= <Header> <PresenationDataSize> <DIB>

<Header> ::= <StandardPresentationObject>

<PresentationDataSize> :: <Integer32>

<DIB> :: = 1*<octet>
```

OFFICE 2003.3.9 END

OFFICE 2003.3.10: Generic Presentation Object

PRODUCT: WORD, EXCEL, POWERPOINT

GRAMMAR:

```
<GenericPresentationObject> :: =
     <StandardClipboardFormatPresentationObject> |
     <RegisteredClipboardFormatPresentationObject>

<StandardClipboardFormatPresentationObject> ::= <Header>
     <PresentationDataSize> <PresentationData>

<Header> :: <ClipboardFormatHeader>

<ClipboardFormatHeader> ::= <PresentationObjectHeader> <Clipboard Format>

<Clipboard Format> ::= <Standard> | <Registered>

<Standard> ::= !0x00000000

<Registered> ::= 0x00000000

<StandardClipboardFormatPresentationObject> ::= <ClipboardFormatHeader>
     <PresentationDataSize> <PresentationData.

<PresentationDataSize ::= <Integer32>

<PresentationData> ::= 1*<octet>

<RegistedClipboardFormatPresentationObject> ::= <ClipboardFormatHeader>
     <StringFormatDataSize> <StringFormatData> <PresentationDataSize>
     <PresentationData>

<StringFromDataSize> :: = <Integer32>

<StringFormatData> ::= <LengthPrefixAnsiString> |
     <LengthPrefixUnicodeString>
```

OFFICE 2003.3.10 END

OFFICE 2003.3.11: OLE 1.0 Object Header

PRODUCT: WORD, EXCEL, POWERPOINTPOWER POINT

GRAMMAR:

```
<OLE_1_0_ObjectHeader> ::= <OLE_1_0_Version> <FormatID> <ClassName>
     <TopicName> <ItemName>

<FormatID> ::= 0x00000001 <!-- Linked Object --> | 0x00000002 <!--
     Embedded Object -->
```

```
<TopicName ::= <LengthPrefixAnsiString>

<ItemName ::= <LengthPrefixAnsiString>
```

OFFICE 2003.3.11 END

OFFICE 2003.3.12: OLE 1.0 Embedded Object

PRODUCT: WORD, EXCEL, POWERPOINT

GRAMMAR:

```
<OLE_1_0_Embedded Object> ::= <OLE_1_0_Header> <NativeDataSize>
     <NativeData>

<NativeDataSize> :: <Integer32>

<NativeData> ::= 1*<octet>
```

OFFICE 2003.3.12 END

OFFICE 2003.3.13: OLE 1.0 Linked Object

PRODUCT: WORD, EXCEL, POWERPOINT

GRAMMAR:

```
<OLE_1_0_Linked Object> :: <OLE_1_0_ObjectHeader> <NetworkName>
     <Reserved> <LinkUpdateOption> <Presentation>

<NetworkName> ::= <LengthPrefixAnsiString>

<Reserved> ::= <Integer32>

<LinkUpdateOption> ::= <Integer32> <!-- Implementation defined hint about
     updating the linked object -->

<Presentation> ::= <MetaFilePresentationObject> | <DIBPresentationObject>
     <BitmapPresentationObject> |
     <StandardClipboardFormatPresentationObject>  |
     <RegisteredClipboardFormatPresentationObject>
```

OFFICE 2003.3.13 END

OLE 2.0 Constructs *(See Section 3 in main document.)*

OFFICE 2003.3.14: ClipboardFormatorAnsiString

PRODUCT: WORD, EXCEL, POWERPOINT

GRAMMAR:

```
ClipboardFormatorAnsiString> ::= <MarkerOrLenght> <FormatorAnsiString>

<MarkerOrLength> ::= <Integer32>

<FormatOrAnsiString: ::= <Standard ClibpboardO Format> | <AnsiString>
```

OFFICE 2003.3.14 END

OFFICE 2003.3.15: ClipboardFformatoOrUnicodeString

PRODUCT: WORD , EXCEL, POWERPOINT

GRAMMAR:

```
<ClipboardFormatorAnsiString> ::= <MarkerOrLenght> <FormatorAnsiString>

<MarkerOrLength> ::= <Integer32>

<FormatOrAnsiString: ::= <Standard ClibpboardO Format> | <AnsiString> |
     Unicode String>
```

OFFICE 2003.3.15 END

OFFICE 2003.3.16: OLE 2.0 Stream

PRODUCT: WORD

GRAMMAR:

```
<OLE_2_0_Embed_Stream> ::= <OLE_2_0_Version> <OLE_2_O_Embed_Flag>
     <OLE_2_0_Common>

<OLE_2_0_Link_Stream> ::=  <OLE_2_0_Vversion>  <OLE_2_0_Link_Flag>
     <OLE_2_0_Common> <Relative Moniker Stream Size> <Relative Moniker
     Stream>  <Absolute Moniker Stream Size> <Absolute Monikekr Stream>
     <ClsidIndicator> <Clsid> <Reserved Display Name> <Reserved2>
```

```
<OLE_2_0_Vversion> ::= 0x02000001

<OLE_2_O_Embed_Flag> ::= 0x00000000

<OLE_2_0_Link_Flag> ::= 0x00000001

<OLE_2_0_Common> ::= <LingUpdateOption> <Reserved1> <Reserved Moniker
     Stream Size> <Reserved Moniker Strea>

<linkUpdownOption> ::= <Integer32>

<Reserved1> ::= 0x00000000 <!-- OLE struct is invalid if not 0x00000000 -
     ->

<Reserved Moniker Stream Size> ::= <Integer32>

<Reserved Moniker Stream> ::= <Moniker Stream>

<Relative Moniker Stream Size> ::= <Integer32>

<Relative Moniker Stream> ::= <Moniker Stream>

<Absolute Moniker Stream Size> ::= <Integer32>

<Absolute Moniker Stream> ::= <Moniker Stream>

<ClsidIndicator> ::= 0xFFFFFFFF <!-- must be a signed 32 bit integer-->

<Clsid> ::= <GUID>

<Reserved Display Name> ::= <LengthPrefixUnicodeString>

<Reserved2>  ::= <Integer3> <!-- must be ingored -->

<Load Update Time> ::= <FILETIME>

<LocalCheckUpdateTime> ::= <FILETIME>

<RemoteUpdateTime> ::= <FILETIME>

<Moniker Stream> ::= <ClsID> <Moniker Stream Data>

<Moniker Stream Data> ::= 1*<octet> <!-- size determined by class id -->
```

OFFICE 2003.3.16 END

OFFICE 2003.3.17: OLE 2.0 Presentation Stream

PRODUCT: WORD, EXCEL, POWERPOINT

GRAMMAR:

```
<OLE_2_0_PresentationStream> ::= <AnsiClipboardFormat> <TargetDeviceSize>
     <TargetDevice> <Aspect> <Lindex> <Advf> <Reserved1> <Width>
     <Height> <Size> <Data> <Reserved2> <TOC Signature>

<AnsiClipboardFormat> ::= <ClipboardFormatorAnsiString>

<TargetDeviceSize> ::= <Integer32>

<TargetDevice> ::= <DVTARGETDEVICE>

<Aspect> ::= <Integer32> <!-- Implementation hint for hot to display data
     --.

<Reserved1> ::=  <Integer32>

<Lindex: ::= <Integer32>

<Advf> ::=  <Integer32>

<Width> ::= <Integer32>

<Height> ::= <Integer32>

<Size> ::= <Integer32>

<Data> :::=0*<Size><octet>

<Reserved2> ::= <octet>[18]

<TOC Data> ::= <empty> | <TOC Signature> <TOC Count> <TOC Entry Array>

<TOC Signature> ::= 0x494E414E

<TOC Count> ::= <Integer16>

<TOC Entry Array> ::= 0*<TOC Count> <TOC Entry> <!-- must follow
     immediately behind the Presentation stream>

<TOC Entry> ::= <AnsiClipboardFormat> <Target Device Size> <Aspect>
     <Lindex> <Tyned> <Reserved3> <Advf> <Reserved4> <Target Device>

<Tyned> ::=<Integer32>

<Reserved3> ::= <Integer32>[3]

<Reserved4> ::= <Integer32>

<DVTTARGETDEVICE> ::= <DriveNameOffset> <DeviceNameOffset>
     <PortNameOffset> <ExDevModeOffset> <Drivername> <DeviceName>
     <PortName> <ExDevMode>

<DriverNameOffset> ::= <Integer16> <!-- offset from start of this
```

```
                    structure to the Driver Name>

               <DevNameOffset> ::= <Integer16> <!-- offset from start of this structure
                    to the Device  Name>

               <PortNameOffset> ::= <Integer16> <!-- offset from start of this structure
                    to the Port Name>

               <ExDevModeOffset> ::= <Integer16> <!-- offset from start of this
                    structure to the ExDevModeName>

               <DriverName> ::= <AnsiString>

               <DeviceName> ::= <AnsiString>

               <PortName> ::= <AnsiString>

               <ExDevMode> ::= <AnsiString>
```

OFFICE 2003.3.17 END

OFFICE 2003.3.18: OLE 2.0 Native Data Stream

PRODUCT: WORD, EXCEL, POWERPOINT

GRAMMAR:

```
        <OLE_2_0_NativeDataStram> ::= <NativeDataSize> <NativeData>

        <NativeDataSize> ::= <Integer32>

        <NativeData> ::= <octet>[<NativeDataSize>]
```

OFFICE 2003.3.18 END

OFFICE 2003.3.19: Compound Object Header

PRODUCT: WORD, EXCEL, POWERPOINT

GRAMMAR:

```
        <Compound Object Header> :: <Reserved1> <Version> <Reserved2?

        <Reserved1> :: <Integer32>

        <Version> ::= <Integer32>
```

```
<Reserved2> ::= <Integer32>[5]
```

OFFICE 2003.3.19 END

OFFICE 2003.3.20: Component Object Stream

PRODUCT: WORD

GRAMMAR:

```
<Component Object Stream> ::=  <Component Object Header> <Ansi User Type>
     <AnsiClipbaordFormat> <Reserved1> <Unicode Marker> <Unicode User
     Type> <Unicode Clipboard Format> <Reserved2>

<Ansi User Type> ::= <LengthPrefixAnsiString>

<AnsiClipboardFormat> ::= <ClipboardFormatOrAnsiString>

<Rreserved1> ::= <empty> | <LengthPrefixAnsiString>

<Unicode Maker> ::= <empty> | 0x71B239F4 <!-- MS Documentation is unclera
     on this; if it is not empty or 0x71B239F4, then it must be ignored.
     However, the data are of variable size, so it is imposible to
     determine the length of the data -->

<Unicode User Type> ::= <LengthPrefixUnicodeString>

<Unicode Clipboard Format> :: <ClipboardFormatOrUnicodeString>

<Rserved2> ::= <LengthPrfixUnicodeString>
```

OFFICE 2003.3.20 END

OFFICE 2003.3.21: Embedded Font Structures (N/A)

OFFICE 2003.3.21 END

File Level Constructs Specific to MS Word *(See Section 3 in main document.)*

OFFICE 2003.3.22: FIB

PRODUCT: WORD

GRAMMAR:

```
<File Information Block> ::= <FIB Base>  <csw> <FibRgW> <cslw> <FibRgL>
     <cbRgFcLcb> <fibRgFcLcBlcok> <cswNew> <fibRgCswMew>

<FIB Base> ::= <wIdent> <nFib> <unused> <lid> <pnNext> <FIB Base Bitmask>
     <nFibBack> <lkey> <envr> <FIB Base Bitmask 2>  <reserved 3>
     <reserved 4> <reserved 5> <reserved 6>

<wIdent> ::= 0xA5EC

<nFib> ::= Integer16 <!-- superseded by fibRgCswNew, should be 0x00C1 -->

<unused> ::= Integer16 <!-- must be ignored -->

<lid> ::= Integer16 <!-- language id -->

<FIB Base Bitmask 1> ::= <A> <B> <C> <D> <E> <F> <G> <H>  <I> <J> <K> <L>
     <M>

<A> ::= bit <!-- fDot, if set document is a template -->

<B> ::= bit <!-- fGlsy, specifies if document only contains AutoText -->

<C: ::= bit <!-- fComplex, specifies whether the last save was an
     incremental save -->

<D> ::= bit <!-- fHasPic, if 0, there should be NO pictures in document -
     ->

<E> ::= bit[4] <!--cQuickSave, may specify the number of incremental
     saves -->

<F> ::= bit <!-- fEncrypted, specifies whether the file is encrypted or
     obfuscated -->

<G> ::= bit <!-- fWhichTblSize, designates the table style>

<H> ::= bit <!-- fReadOnlyRecommended, specifies whether author specified
     the document should be opened read only -->

<I> ::= bit <!-- fWriteReservation, specifies whether the document has a
     write reservation password -->

<J> ::= bit <!-- fExCha, must be set to 1 -->

<K> ::= bit <!-- fLoadOverride, specifies whether to override the normal
     style's language and font with the default language and font -->
```

```
<L> ::= bit <!-- fEastAsian, specifies whether the installed language of
     the application is an East Asian language -->

<M> ::= bit <!-- fObfuscated, if fEncrypted is 1, specifies whether the
     document is obfuscated -->

<nFibBack> ::= Integer16 <!-- Should be 0x00BF, must be 0x00BF or 0x000C1
     -->

<lkey> ::= Integer32 <!-- if fEncryption = 1 and fObfuscation = 1,
     specifies the XOR verifier for the obfuscation; if fEncryption =1
     and fObfuscation = 0, specifies the size of the Encryption header;
     otherwise must be 0 -->

<envr> :: octet  <!-- must be 0 and must be ignored -->

<FIB Base Bitmask 2> ::= <N> <0> <P> <Q> <R> <S>

<N> ::= bit <!-- fMac, must be 0 and must be ignored -->

<O> ::= bit <!-- fEncryptSpectial, should be 0; should be ignored -->

<P> ::= bit <!-- fLoadOverridePage, specifies whether to override the
     page layout characteristics with the default appropriate for the
     language and application -->

<Q> ::= bit <!-- reserved 1; undefined, must be ignored -->

<R> ::= bit <!-- reserved 2; undefined, must be ignored -->

<S> ::= bit[3]  <!-- fSpare, undefined, must be ignored -->

<reserved 3> ::= Integer16 <!-- must be 0, must be ignored -->

<reserved 4> ::= Integer16 <!-- must be 0, must be ignored -->

<reserved 5> ::= Integer32 <!-- undefined, must be ignored -->

<reserved 6> ::= Integer32 <!-- undefined, must be ignored -->

<csw> :: Integer16 <!-- represents the 16 bi values corresponding to the
     fibRgW that follows; must be 0x000e -->

<fibRgW> :: <reserved space> <lidFE> <!-- 28 bytes; the flRgW97 -->

<reserved space> ::= Integer16[13] <!-- essentially an array of
     Integer16, first 4 are undefined and must be ignored; remaining 9
     must be 0 and must be ignored -->

<lid> :::= Integer16 <!-- language id; dependent on the nFib value. If
     0x00c1 and if the FIB fFarEast value is true (1), then this is the
     lid of the stored style names. If nFib equal to 0x00D9, 0x0101,
     0x010C, or 0x0112 ; this represents the lid of the stored style
     names -->
```

<cslw> :: Integer16 <!-- Represents the count of 32 bit values in the flbRgLw that follows. Must be 0x0015 -->

<flRgLw> ::= Integer32[22] <!-- the flRglw97. In the interest of space, the reader is referred to the MS-DOS.pdf, section 2.5.4 for a full discussion of the structure and its contents -->

<cbRgFcLob> ::= Integer16 <!-- Represents the count of 65 bit values in the fibRgFcLcbBlob that follows. Based on the value of the nFib -->

nFib	cbRgFcLob
0x00c1	0x005d
0x00D9	0x006C
0x0101	0x0088
0x010C	0x00A4
0x0112	0x00B7

fibRgFcLcbBlob ::= <!-- The FibRgFcLc. In the interest of space, the reader is referred to the MS-DOS.pdf, section 2.5.5 for a full discussion of the structure and its contents -->

cswnew ::= Integer16 <!-- Represents the count of 16 bit values corresponding to the fubRgCswNew below; based on the value of the nFib -->

nFib	cbRgFcLob
0x00c1	0
0x00D9	0
0x0101	0x00002
0x010C	0x00002
0x0112	0x00005

fibRgCswNew ::= <! If cswNew is non zero, this is the fibRGCswNew. In the interest of space, the reader is referred to MS-DOS.pdf, section 2.5.11, for a detailed treatment of this structure.

OFFICE 2003.3.22 END

File Level Constructs Specific to MS Excel *(See Section 3 in main document.)*

```
<Excel Record Start> ::= <Record Tag> <Record Length>

<Record Tag> ::= Integer16

<Record Length ::= Integer16

<Obj> ::= <FtCmo> <FTGmo> <FtCf> <FtPioGrbit> <FtCbls> <FtRbo> <FtSbs>
     <FtNts> <FtMacro> <FtPictFmla> <ObjLinkFmla> <FtCblsData>
     <FtRboData> <FtEdoData> <FtLbsData> <FtGboData> <octet-zero-
     ignored>[4]

<FtCmo> ::= <0x15> <0x12> <ot> <id> <fLocked> <bit-zero> <fDefaultSize>
     <fPublished> <fPrint> <bit-ignored> <bit-ignored> <fDisabled>
     <fUIObj> <fRecalcObj> <bit-ignored> <bit-ignored>
     <fRecalcObjAlways> <bit-ignored> <bit-ignored> <bit-ignored>
     <octet-ignored>[4] <octet-ignored>[4] <octet-ignored>[4]

<ot> ::= <Group> | <Line> | <Rectangle> | <Oval> | <Arc> | <Chart> |
     <Text> | <Button> | <Picture> | <Polygon> | <Checkbox> | <Radio
     Button> | <Edit Box> | <Labal> | <Dialog Box> | <Spin control> |
     <Scrollbar> | <List> | <Group Box> | <Dropdown list> | <Note> |
     OfficeArt object>

<id> ::= <octet>[2]

<fLocked> ::= <bit>

<fDefaultSize> ::= <bit>

<fPublished> ::= <bit>

<fPrint> ::= <bit>

<fDisabled> ::= <bit>

<fUIObj> ::= <bit>

<fRecalcObj> ::= <bit>

<fRecalcOjbAlways> ::= <bit>

<XLUnicodeString> ::= <cch> <fHighByte> <reserved> <rgb>

<XLUnicodeStringNoCch> ::= <fHighByte> <reserved> <rgb>

<cch> ::= <Integer16>
```

```
<fHighByte> ::= <bit>

<reserved> ::= <bit>[7]

<rgb> ::= <octet>[cch]
```

OFFICE 2003.3.23: General Excel Record

PRODUCT: EXCEL

GRAMMAR:

```
<Record> ::= <Excel Record Start> <Record Data>

<Record Data> :: = 1*8224<octet>
```

OFFICE 2003.3.23 END

OFFICE 2003.3.24: Beginning of File (BOF) Record

PRODUCT: EXCEL

GRAMMAR:

```
<BOF Record> ::= <Excel Record Start> <BOF>

<BOF> ::= <version> <document type> <build id> <BIFF year> <BOF Bit Mask
     1> <BOF Bit Mask 2>

<version> ::= Integer16 <!-- must be 0x0600 -->

<document type ::= <wb ss> | <di ss> | <ch ss> | <mac ss>

<wb ss> ::= 0x0005 <!-- workbook sub-stream -->

<di ss> ::= 0x0010 <!-- dialog sheet substream or worhseet substream -->

<ch ss> ::= 0x0020 <!-- the chart substream -->

<mac ss> ::= 0x0040 <!-- the macro substream -->

<build id> ::= Integer16 <!-- build identifier -->

<BIFF Year ::= Integer16 <!-- must be 0x07CC or 0x07CD -->

<BOF Bitmask 1> :: = <A> <B> <C> <D> <E> <F> <G> <H> <I> <J> <K> <L> <M>
     <N>
```

<A> :: bit <!-- fWin: A bit that specifies whether this file was last edited on a Windows platform. The value MUST be 1 -->

<B> :: = bit <!-- fRisc: A bit that specifies whether the file was last edited on a RISC platform. The value MUST be 0. -->

<C> ::= bit <!-- fBeta (1 bit): A bit that specifies whether this file was last edited by a beta version of the application. The value MUST be 0. -->

<D> ::= bit <!-- fWinAny (1 bit): A bit that specifies whether this file has ever been edited on a Windows platform. The value SHOULD <28> be 1.

<E> ::= bit <!-- fMacAny (1 bit): A bit that specifies whether this file has ever been edited on a Macintosh platform. The value MUST be 0. -->

<F> ::= bit <!-- fBetaAny (1 bit): A bit that specifies whether this file has ever been edited by a beta version of the application. The value MUST be 0. -->

<G> ::= bit <!-- unused1 (2 bits): Undefined and MUST be ignored. -->

<H> ::= bit <!-- fRiscAny (1 bit): A bit that specifies whether this file has ever been edited on a RISC platform. The value MUST be 0. -->

<I> ::= bit <!-- fOOM (1 bit): A bit that specifies whether this file had an out-of-memory failure. -->

<J> ::= bit <!-- fGlJmp (1 bit): A bit that specifies whether this file had an out-of-memory failure during rendering. -->

<K> ::= bit <!-- unused2 (2 bits): Undefined, and MUST be ignored. -->

<L> ::= bit <!-- fFontLimit (1 bit): A bit that specified whether this file hit the 255 font limit <29>. -->

<M> ::= bit <!-- verXLHigh (4 bits): An unsigned integer that specifies the highest version of the application that once saved this file. MUST be a value from the following table:

Value	Meaning
0x0	Specifies the highest version of the application that has ever saved this file. <30>
0x1	Specifies the highest version of the application that has ever saved this file. <31>
0x2	Specifies the highest version of the application that has ever saved this file. <32>

0x3	Specifies the highest version of the application that has ever saved this file. <33>
0x4	Specifies the highest version of the application that has ever saved this file. <34>
0x6	Specifies the highest version of the application that has ever saved this file. <35>

```
-->

<N> ::= bit <!-- unused3 (1 bit): Undefined, and MUST be ignored. -->

<reserved1> ::= bit[13] <!--  MUST be zero, and MUST be ignored. -->
```

OFFICE 2003.3.24 END

OFFICE 2003.3.25: Future Records

PRODUCT: EXCEL

GRAMMAR:

```
<Future Record Header> ::=  <FRTHeader> |  <FrtHeaderOld> |
     <FrtRefHeader> | <FrtRefHeaderNoGrBit>  |  <FrtRefHeaderU>

<FrtHeader> :: = <rt> <frtflags>

<rt> ::= Integer16 <!-- record type. MUST match the record type in the
     general information preceding this record -->

<frtflags> ::= <A> <B> <reserved>

<A> ::= <frtRef>

<frtref ::= bit <!-- indicate if the containing record is a range of
     cells:  0 inidcate it does not. 1 means it does. -->

<B> ::= <frtAlert>

<frtAlert ::= bit <!-- indicates whether to warn the user when saving the
     file that this record was not recgnized -->

<reserved> ::= octet[14] <!-- must be zero and must be ignored -->

<FrtHeaderOld> ::= <rt> <frtHader> <!-- frtHeader.frtflags.frtRef must be
```

```
                0; frtHeader.frtflags.frtAlert must be 0 -->

          <FrtRefHeader> ::= <rt> <frtflags> <!-- frtAlert must be 0 --> <ref8>

          <ref8> ::= <rwFirst> <rwLast> <colFirst> <colLast>

          <rwFirst> ::= <rwx>

          <rwLast> ::= <rwx>

          <colFirst> ::= <colx>

          <colLast> ::= <colx>

          <rwx> ::= Integer16 <!-- row reference -->

          <colx> ::= Integer16 <!-- column refernce -->

          <FrtRefHeaderNoGrBit>  ::= <rt> <ref8>

          <FrtRefHeaderU> ::= <rt> <frtflags> <ref8>
```

OFFICE 2003.3.25 END

File Level Constructs Specific to MS PowerPoint *(See Section 3 in main document.)*

PowerPoint Record Types

```
          <RecordType> ::= <RT_Document> | <RT_DocumentAtom> | <RT_EndDocumentAtom>
               | <RT_Slide> | <RT_SlideAtom> | <RT_Notes> | <RT_NotesAtom> |
               <RT_Environment> | <RT_SlidePersistAtom> | <RT_MainMaster> |
               <RT_SlideShowSlideInfoAtom> | <RT_SlideViewInfo> | <RT_GuideAtom> |
               <RT_ViewInfoAtom> | <RT_SlideViewInfoAtom> | <RT_VbaInfo> |
               <RT_VbaInfoAtom> | <RT_SlideShowDocInfoAtom> | <RT_Summary> |
               <RT_DocRoutingSlipAtom> | <RT_OutlineViewInfo> |
               <RT_SorterViewInfo> | <RT_ExternalObjectList> |
               <RT_ExternalObjectListAtom> | <RT_DrawingGroup> | <RT_Drawing> |
               <RT_GridSpacing10Atom> | <RT_RoundTripTheme12Atom> |
               <RT_RoundTripColorMapping12Atom> | <RT_NamedShows> | <RT_NamedShow>
               | <RT_NamedShowSlidesAtom> | <RT_NotesTextViewInfo9> |
               <RT_NormalViewSetInfo9> | <RT_NormalViewSetInfo9Atom> |
               <RT_RoundTripOriginalMainMasterId12Atom> |
               <RT_RoundTripCompositeMasterId12Atom> |
               <RT_RoundTripContentMasterInfo12Atom> | <RT_RoundTripShapeId12Atom>
               | <RT_RoundTripHFPlaceholder12Atom> |
               <RT_RoundTripContentMasterId12Atom> |
               <RT_RoundTripOArtTextStyles12Atom> |
               <RT_RoundTripHeaderFooterDefaults12Atom> |
               <RT_RoundTripDocFlags12Atom> |
               <RT_RoundTripShapeCheckSumForCL12Atom> |
```

```
                    <RT_RoundTripNotesMasterTextStyles12Atom> |
                    <RT_RoundTripCustomTableStyles12Atom> | <RT_List> |
                    <RT_FontCollection> | <RT_FontCollection10> |
                    <RT_BookmarkCollection> | <RT_SoundCollection> |
                    <RT_SoundCollectionAtom> | <RT_Sound> | <RT_SoundDataBlob> |
                    <RT_BookmarkSeedAtom> | <RT_ColorSchemeAtom> | <RT_BlipCollection9>
                    | <RT_BlipEntity9Atom> | <RT_ExternalObjectRefAtom> |
                    <RT_PlaceholderAtom> | <RT_ShapeAtom> | <RT_ShapeFlags10Atom> |
                    <RT_RoundTripNewPlaceholderId12Atom> | <RT_OutlineTextRefAtom> |
                    <RT_TextHeaderAtom> | <RT_TextCharsAtom> | <RT_StyleTextPropAtom> |
                    <RT_MasterTextPropAtom> | <RT_TextMasterStyleAtom> |
                    <RT_TextCharFormatExceptionAtom> |
                    <RT_TextParagraphFormatExceptionAtom> | <RT_TextRulerAtom> |
                    <RT_TextBookmarkAtom> | <RT_TextBytesAtom> |
                    <RT_TextSpecialInfoDefaultAtom> | <RT_TextSpecialInfoAtom> |
                    <RT_DefaultRulerAtom> | <RT_StyleTextProp9Atom> |
                    <RT_TextMasterStyle9Atom> | <RT_OutlineTextProps9> |
                    <RT_OutlineTextPropsHeader9Atom> | <RT_TextDefaults9Atom> |
                    <RT_StyleTextProp10Atom> | <RT_TextMasterStyle10Atom> |
                    <RT_OutlineTextProps10> | <RT_TextDefaults10Atom> |
                    <RT_OutlineTextProps11> | <RT_StyleTextProp11Atom> |
                    <RT_FontEntityAtom> | <RT_FontEmbedDataBlob> | <RT_CString> |
                    <RT_MetaFile> | <RT_ExternalOleObjectAtom> | <RT_Kinsoku> |
                    <RT_Handout> | <RT_ExternalOleEmbed> | <RT_ExternalOleEmbedAtom> |
                    <RT_ExternalOleLink> | <RT_BookmarkEntityAtom> |
                    <RT_ExternalOleLinkAtom> | <RT_KinsokuAtom> |
                    <RT_ExternalHyperlinkAtom> | <RT_ExternalHyperlink> |
                    <RT_SlideNumberMetaCharAtom> | <RT_HeadersFooters> |
                    <RT_HeadersFootersAtom> | <RT_TextInteractiveInfoAtom> |
                    <RT_ExternalHyperlink9> | <RT_RecolorInfoAtom> |
                    <RT_ExternalOleControl> | <RT_SlideListWithText> |
                    <RT_AnimationInfoAtom> | <RT_InteractiveInfo> |
                    <RT_InteractiveInfoAtom> | <RT_UserEditAtom> | <RT_CurrentUserAtom>
                    | <RT_DateTimeMetaCharAtom> | <RT_GenericDateMetaCharAtom> |
                    <RT_HeaderMetaCharAtom> | <RT_FooterMetaCharAtom> |
                    <RT_ExternalOleControlAtom> | <RT_ExternalMediaAtom> |
                    <RT_ExternalVideo> | <RT_ExternalAviMovie> | <RT_ExternalMciMovie>
                    | <RT_ExternalMidiAudio> | <RT_ExternalCdAudio> |
                    <RT_ExternalWavAudioEmbedded> | <RT_ExternalWavAudioLink> |
                    <RT_ExternalOleObjectStg> | <RT_ExternalCdAudioAtom> |
                    <RT_ExternalWavAudioEmbeddedAtom> | <RT_AnimationInfo> |
                    <RT_RtfDateTimeMetaCharAtom> | <RT_ExternalHyperlinkFlagsAtom> |
                    <RT_ProgTags> | <RT_ProgStringTag> | <RT_ProgBinaryTag> |
                    <RT_BinaryTagDataBlob> | <RT_PrintOptionsAtom> |
                    <RT_PersistDirectoryAtom> | <RT_PresentationAdvisorFlags9Atom> |
                    <RT_HtmlDocInfo9Atom> | <RT_HtmlPublishInfoAtom> |
                    <RT_HtmlPublishInfo9> | <RT_BroadcastDocInfo9> |
                    <RT_BroadcastDocInfo9Atom> | <RT_EnvelopeFlags9Atom> |
                    <RT_EnvelopeData9Atom> | <RT_MacQTTransitionInfoAtom> |
                    <RT_HashCodeAtom> | <RT_VisualPageAtom> | <RT_BuildList> |
                    <RT_BuildAtom> | <RT_ChartBuild> | <RT_ChartBuildAtom> |
                    <RT_DiagramBuild> | <RT_DiagramBuildAtom> | <RT_ParaBuild> |
                    <RT_ParaBuildAtom> | <RT_LevelInfoAtom> |
                    <RT_RoundTripAnimationAtom12Atom> |
                    <RT_AnimationHashAtom12Deprecated> |
                    <RT_RoundTripAnimationHashAtom12Atom> | <RT_Comment10> |
                    <RT_Comment10Atom> | <RT_CommentIndex10> | <RT_CommentIndex10Atom>
                    | <RT_LinkedShape10Atom> | <RT_LinkedSlide10Atom> |
                    <RT_SlideFlags10Atom> | <RT_SlideTime10Atom> | <RT_DiffTree10> |
```

```
                    <RT_Diff10> | <RT_Diff10Atom> | <RT_SlideListTableSize10Atom> |
                    <RT_SlideListEntry10Atom> | <RT_SlideListTable10> |
                    <RT_CryptSession10Container> | <RT_FontEmbedFlags10Atom> |
                    <RT_FilterPrivacyFlags10Atom> | <RT_DocToolbarStates10Atom> |
                    <RT_PhotoAlbumInfo10Atom> | <RT_SmartTagStore11Container> |
                    <RT_RoundTripSlideSyncInfo12> | <RT_RoundTripSlideSyncInfoAtom12> |
                    <RT_TimeConditionContainer> | <RT_TimeNode> | <RT_TimeCondition> |
                    <RT_TimeModifier> | <RT_TimeBehaviorContainer> |
                    <RT_TimeAnimateBehaviorContainer> | <RT_TimeColorBehaviorContainer>
                    | <RT_TimeEffectBehaviorContainer> |
                    <RT_TimeMotionBehaviorContainer> |
                    <RT_TimeRotationBehaviorContainer> |
                    <RT_TimeScaleBehaviorContainer> | <RT_TimeSetBehaviorContainer> |
                    <RT_TimeCommandBehaviorContainer> | <RT_TimeBehavior> |
                    <RT_TimeAnimateBehavior> | <RT_TimeColorBehavior> |
                    <RT_TimeEffectBehavior> | <RT_TimeMotionBehavior> |
                    <RT_TimeRotationBehavior> | <RT_TimeScaleBehavior> |
                    <RT_TimeSetBehavior> | <RT_TimeCommandBehavior> |
                    <RT_TimeClientVisualElement> | <RT_TimePropertyList> |
                    <RT_TimeVariantList> | <RT_TimeAnimationValueList> |
                    <RT_TimeIterateData> | <RT_TimeSequenceData> | <RT_TimeVariant> |
                    <RT_TimeAnimationValue> | <RT_TimeExtTimeNodeContainer> |
                    <RT_TimeSlaveContainer>

               <PrintableUnicodeString> ::= <UTF-16 Unicode String> <!-- but not 0x0000
                    – 0x001F, 0x007F – 0x009F -->

               <TabCrLfPrintableUnicodeString> ::= <UTF-16 Unicode String> <!-- but not
                    0x0000 – 0x0008, 0x000B, 0x000C, 0x000e – 0x001F, 0x007F – 0x009F -
                    ->

               <PrintableAnsiString> ::= <AnsiStringNullTerminated> <!-- but not 0x00 –
                    0x1F, 0x7F – 0x9F -->

               <FileOrDirNameFragment> ::= <UTF-16 Unicode String>
```

REFERENCE:
See [MS-PPT], Section 2.13.24 RecordType

Office 2003 Constructs and Metadata *(See Section 3 in main document.)*

OFFICE 2003.4.1: Field Codes

PRODUCT: WORD

GRAMMAR:

```
               <Plcfld> ::= *<CP> *<Fld>

               <CP> ::= <!-- A character position, CP, is an unsigned 32-bit integer
```

```
                that serves as the zero-based index of a character in the document
                text. [MS-DOC].pdf - Section 2.2.1 -->

          <Fld> ::= <fldch> <grffld>

          <fldch> ::=  0x13 <A> | 0x14 <A> | 0x15 <A>   <!-- This value controls
                the implementation of grffld. A value of 0x13 indicates grffld uses
                flt, 0x14 indicates grffld is null and 0x15 indicates grffld is
                grffldEnd. -->

          <A> ::= <!-- Three reserved  bits, which an application must ignore. -->

          <grffld> ::= <flt> | null | <grffldEnd>

          <flt> ::= <!-- An index to a field type. Field type values are enumerated
                in [MS-DOC].pdf - Section 2.9.90 -->
```

OFFICE 2003.4.1: END

OFFICE 2003.4.2: Macros

PRODUCT: WORD, EXCEL, POWERPOINT

GRAMMAR:

```
          <Project Root Storage> ::= <VBA> *<Designer Storage> <PROJECTlk>
                <PROJECTwm> <PROJECT>

          <VBA> ::= <_VBA_PROJECT> <dir> *<Module Stream> *<__SRP_n>

          <Designer Storage> ::= <VBFrame Stream> <Form Designer owned storages and
                streams>
```

OFFICE 2003.4.2: END

OFFICE 2003.4.3: Comments

PRODUCT: WORD

GRAMMAR:

```
          <PlcfandTxt> ::= <aCP>

          <aCP> ::= <CP> | <CP> <aCP>

          <CP> ::= <Integer32> <!--Character Position: Specifies positions of
```

```
        comments within the comment document -->

<PlcfandRef> ::= <aCP2> <aATRDPre10>

<aCP2> ::= <CP2> | <CP2> <aCP2>

<CP2> ::= <Integer32> <!--Character Position: Specifies positions of
     comments within the main document -->

<aATRDPre10> ::= <ATRDPre10> | <ATRDPre10> <aATRDPre10>

<ATRDPre10> ::= <xstUsrInitl> <ibst> <bitsNotUsed> <grfNotused>
     <ITagBkmk>

<xstUsrInitl> ::= <octet>[20]

<ibst> ::= <octet>[2]

<bitsNotUsed> ::= <octet-zero-ignored>[2]

<grfNotused> ::= <octet-zero-ignored>[2]

<ITagBkmk> ::= <octet>[4]
```

PRODUCT: EXCEL

GRAMMAR:

```
<NoteRR> ::= <RRD> <bitDelNote> <bitAddNote> <reserved1> <RwU> <ColU>
     <reserved2> <fShow> <reserved3> <fRwHidden> <fColHidden>
     <reserved4> <unused1> <reserved5> <GUID> <ichEnd> <cchNote>
     <stAuthor> <unused2>

<bitDelNote> ::= <bit>

<bitAddNote> ::= <bit>

<reserved1> ::= <bit-zero-ignored>[14]

<RwU> ::= <Integer16>

<ColU> ::= <Integer16>

<reserved2> ::= <bit-zero-ignored>

<fShow> ::= <bit>

<reserved3> ::= <bit-zero-ignored>[5]

<fRwHidden> ::= <bit>

<fColHidden> ::= <bit>

<reserved4> ::= <bit-zero-ignored>[2]
```

```
<unused1> ::= <bit-ignored>

<reserved5> ::= <bit-zero-ignored>[4]

<ichEnd> ::= <Integer32>

<cchNote> ::= <Integer32>

<stAuthor> ::= <XLUnicodeString>

<unused2> ::= <octet-ignored>[2]

<NoteSh> ::= <RW> <Col> <reserved1> <fShow> <reserved2> <unused1>
     <reserved3> <fRwHidden> <fColHidden> <reserved4> <ObjId> <stAuthor>
     <unused2>

<RW> ::= <Integer16>

<Col> ::= <Integer16>

<reserved1> ::= <bit-zero-ignored>

<fShow> ::= <bit>

<reserved2> ::= <bit-zero-ignored>

<unused1> ::= <bit-ignored>

<reserved3> ::= <bit-zero-ignored>[3]

<fRwHidden> ::= <bit>

<fColHidden> ::= <bit>

<reserved4> ::= <bit-zero-ignored>[7]

<ObjId> ::= <octet>[2]

<stAuthor> ::= <XLUnicodeString>

<unused2> ::= <octet-ignored>

<FtNts> ::= <0x000D> <0x0016> <GUID> <octet-boolean>[2] <octet-
     ignored>[4] <!-- FtNts is a structure used in an Obj when the enum
     “ot” indicates a note.  Referenced by the ObjId.  See Obj in the
     Excel general constructs.  FtNts is specifically used for
     notes/comments -->
```

PRODUCT: POWERPOINT

GRAMMAR:

```
<Comment10Container> ::= <rh1> <Comment10AuthorAtom> <Comment10TextAtom>
     <Comment10AuthorInitialAtom> <Comment10Atom>
```

```
<rh1> ::= <0xF> <0x000> <RT_Comment10>

<Comment10AuthorAtom> ::= <rh2> <PrintableUnicodeString>

<rh2> ::= <0x0> <0x000> <RT_CString> <rh.recLen> <!--rh.recLen must be an
     even number. It MUST be less than or equal to 104-->

<Comment10TextAtom> ::= <rh3> <TabCrLfPrintableUnicodeString>

<rh3> ::= <0x0> <0x001> <RT_CString> <rh.recLen> <!--rh.recLen must be an
     even number. It MUST be less than or equal to 64000-->

<Comment10AuthorInitialAtom> ::= <rh4> <PrintableUnicodeString>

<rh4> ::= <0x0> <0x002> <RT_CString> <rh.recLen> <!--rh.recLen must be an
     even number. It MUST be less than or equal to 104-->

<Comment10Atom> ::= <rh5> <Integer32> <DateTimeStruct> <PointStruct>

<DateTimeStruct> ::= <octet>[16] <!-- a SYSTEMTIME structure-->

<PointStruct> ::= <Integer32> <Integer32> <!-- x,y coordinates -->

<rh5> ::= <0x0> <0x000> <RT_Comment10Atom> <0x0000001C>
```

OFFICE 2003.4.3: END

OFFICE 2003.4.4: Save History

PRODUCT: WORD

GRAMMAR:

```
<SttbSavedBy> ::= <fExtend> <cData> <cbExtra> <Data>

<fExtend> ::= <0xFFFF>

<Data> ::= <cchData> <data> <extraData> | <cchData> <data> <extraData>
     <Data>

<cData> ::= <octet>[2] <!-- must be less than or equal to 0x0014 -->

<cbExtra> ::= <octet-zero>[2]

<cchData> ::= <octet> | <octet>[2] <!-- based on fExtend -->

<data> ::= <octet>[2*cchData] | <octet>[cchData] <!--variable size -->

<extraData> ::= <octet>[cbExtra]
```

OFFICE 2003.4.4: END

OFFICE 2003.4.5: Template Name

PRODUCT: WORD

LOCATION:
The **SttbfAssoc** record is based on a **STTB** structure. It maintains the file path of the template file via the value at index **0x01**.

GRAMMAR:

```
<SttbfAssoc> ::= <fExtend> <cData> <cbExtra> <Data>

<fExtend> ::= <0xFFFF>

<Data> ::= <cchData> <data> <extraData> | <cchData> <data4> <extraData>
     <Data>

<cData> ::= <0x0012>

<cbExtra> ::= <octet-zero>[2]

<cchData> ::= <octet> | <octet>[2] <!-- based on fExtend -->

<data> ::= <octet>[2*cchData] | <octet>[cchData] <!--variable size -->

<extraData> ::= <octet>[cbExtra]
```

OFFICE 2003.4.5: END

OFFICE 2003.4.6: Scenarios

PRODUCT: EXCEL

GRAMMAR:

```
<ScenMan> ::= <csct> <isctCur> <isctShown> <irefResult> <rgref>

<csct> ::= <Integer16>

<isctCur> ::= <Integer16>

<isctShown> ::= <Integer16>

<irefResult> ::= <Integer16> <!-- range 0-32 -->
```

```
<rgref> ::= <Ref8U> | <Ref8U> <rgref>

<Scenario> ::= <cref> <fLocked> <fHidden> <cchName> <cchComment>
     <cchNameUser> <rgchName> <rgchNameUser> <rgchComment> <rgSLC>
     <rgst> <unused2>

<cref> ::= <Integer16>

<fLocked> ::= <octet-boolean>

<fHidden> ::= <octet-boolean>

<cchName> ::= <Integer8>

<cchComment> ::= <Integer8>

<cchNameUser> ::= <Integer8>

<rgchName> ::= <XLUnicodeStringNoCch>

<rgchNameUser> ::= <XLUnicodeString> <!-- size <= 52 -->

<rgchNameComment> ::= <XLUnicodeString>

<rgSLC> ::= <SLCO8> | <SLCO8> <rgSLC>

<SLCO8> ::= <RwU> <ColSlco8U>

<RwU> ::= <Integer16>

<ColSlco8U> ::= <octet>[2]

<rgst> ::= <XLUnicodeString> | <XLUnicodeString> <rgst>

<unused2> ::= <octet>[2*cref]
```

OFFICE 2003.4.6: END

OFFICE 2003.4.7: Hyperlinks

PRODUCT: WORD, EXCEL, POWERPOINT

GRAMMAR:

```
<VtHyperlinks> ::= <wType> <padding> <VtHyperLinkValue>

<wType> ::= <!-- An unsigned integer that MUST be equal to VT_BLOB
     (0x0041). -->

<padding> ::= <!-- An unsigned integer that MUST be 0x0000. -->
```

```
<VtHyperLinkValue> ::= <cbData> <VecVtHyperlink>

<cbData> ::= <!-- An unsigned integer that specifies the size in bytes of
     VecVtHyperlink-->

<VecVtHyperlink> ::= <cElements> *<VtHyperlink>

<cElements> ::= <!-- An unsigned integer specifying the count of elements
     in the rgHyperlink field. The number of elements in rgHyperlink
     MUST be 1/6 of this value. This value MUST be evenly divisible by
     6. -->

<VtHyperlink> ::= <dwHash> <dwApp> <dwOfficeArt> <dwInfo> <hlink1>
     <hlink2>

<dwHash> ::= <!-- MUST be a VT_I4 TypedPropertyValue as specified in [MS-
     OLEPS] section 2.15. The value of this structure should be
     calculated as specified in the Hyperlink Hash section given hlink1
     and hlink2 field string values as input. -->

<dwApp> ::= <!-- MUST be a VT_I4 TypedPropertyValue as specified in [MS-
     OLEPS] section 2.15. The value of this structure is implementation
     specific. -->

<dwOfficeArt> ::= <!-- MUST be a VT_I4 TypedPropertyValue as specified in
     [MS-OLEPS] section 2.15. The value of this structure MUST be
     MSOSPID type value ([MS-ODRAW] section 2.1.2) specifying the
     indetifier of the shape ([MS-ODRAW] section 2.1.31) to which this
     hyperlink applies in the document. If this hyperlink does not apply
     to a shape, the value field of this structure MUST be 0x00000000. -
     ->

<dwInfo> ::= <!-- MUST be a VT_I4 TypedPropertyValue as specified in [MS-
     OLEPS] section 2.15. The value of this structure is implementation
     specific. The high order 2-byte integer of the value field of this
     structure should be 0x0000. -->

<hlink1> ::= <!-- MUST be a VtString structure with hlink1.wType equal to
     VT_LPWSTR. hlink1.stringValue specifies the hyperlink target. -->

<hlink2> ::= <!-- MUST be a VtString structure with hlink2.wType equal to
     VT_LPWSTR. hlink2.stringValue specifies the hyperlink location. -->
```

OFFICE 2003.4.7: END

OFFICE 2003.4.8: Summary Properties

PRODUCT: WORD, EXCEL, POWERPOINT

GRAMMAR:

```
<PropertySetStream> ::= <PropertySet>+

<PropertySet> ::= <PropertyIdentifierAndOffset>+ <Property>+

<PropertyIdentifierAndOffset> ::= <PropertyIdentifier> <Offset>

<PropertyIdentifier> ::= <!-- Predefined list outlined in [MS-OLEPS].pdf
     - Section 2.1 -->

<Offset> ::= <!-- An integer representing the offset in bytes from the
     beginning of the PropertySet packet to the beginning of the
     Property field -->

<Property> ::= <TypedPropertyValue> | <Dictionary> <!-- Dictionary is
     described in the following Custom Properties section -->

<TypedPropertyValue> ::= <Type> <Padding> <Value>

<Type> ::= <PropertyType>

<PropertyType> ::= <!-- Predefined list outlined in [MS-OLEPS].pdf -
     Section 2.2 -->

<Padding> ::= “0” <!-- Must be set to zero -->

<Value> ::= <!-- Must be the value of the property represented and
     serialized according to the value of Type -->
```

OFFICE 2003.4.8: END

OFFICE 2003.4.9: Custom Properties

PRODUCT: WORD, EXCEL AND POWERPOINT

GRAMMAR:

```
<PropertySetStream> ::= *<PropertySet>

<PropertySet> ::= *<PropertyIdentifierAndOffset> *<Property>

<PropertyIdentifierAndOffset> ::= <PropertyIdentifier> <Offset>

<PropertyIdentifier> ::= <!-- Predefined list outlined in [MS-OLEPS].pdf
     - Section 2.1 --> <Normal> | <DICTIONARY_PROPERTY_IDENTIFIER> |
     <CODEPAGE_PROPERTY_IDENTIFIER> | <LOCALE_PROPERTY_IDENTIFIER> |
     <BEHAVIOR_PROPERTY_IDENTIFIER>

<Normal> ::= 0x00000002 – 0x7FFFFFFF  <!—- The MS documentation specifies
     the following four tags with the same value. We include this for
     completeness and should be revised when the actual values are
     determined. -->

<DICTIONARY_PROPERTY_IDENTIFIER> ::= 0x00000000

<CODEPAGE_PROPERTY_IDENTIFIER> ::= 0x00000000

<LOCALE_PROPERTY_IDENTIFIER> ::= 0x00000000

<BEHAVIOR_PROPERTY_IDENTIFIER> ::= 0x00000000

<Offset> ::= <HEX_INTEGER>  <!-- An integer representing the offset in
     bytes from the beginning of the PropertySet packet to the beginning
     of the Property field -->

<HEX_INTEGER> ::= <HEX_BYTE> <HEX_BYTE> <HEX_BYTE> <HEX_BYTE>

<HEX_BYTE ::= <HEX_NUM> <HEX_NUM>

<HEX_NUM> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E |
     F

<Property> ::= <TypedPropertyValue> | <Dictionary>

<TypedPropertyValue> ::= <Type> <Padding> <Value>

<Type> ::= <PropertyType>

<PropertyType> ::=  <!-- Predefined list outlined in [MS-OLEPS].pdf -
     Section 2.2 --> <VT_EMPTY> | <VT_NULL> | <VT_I2> | <VT_I4> |
     <VT_R4> | <VT_R8> | <VT_CY> | <VT_DATE> | <VT_BSTR> | <VT_ERROR> |
     <VT_BOOL> | <VT_DECIMAL> | <VT_I1> | <VT_UI1> | <VT_UI2> | <VT_UI4>
     | <VT_I8> | <VT_UI8> | <VT_INT> | <VT_UINT> | <VT_LPSTR> |
     <VT_LPWSTR> | <VT_FILETIME> | <VT_BLOB> | <VT_STREAM> |
     <VT_STORAGE> | <VT_STREAMED_Object> | <VT_STORED_Object> |
     <VT_BLOB_Object> | <VT_CF> | <VT_CLSID> | [ <VT_VERSIONED_STREAM> ]
```

```
                    | [ <VT_VECTOR_VT_I2> ] | [ <VT_VECTOR_VT_I4> ] | [
                    <VT_VECTOR_VT_R4> ] | [ <VT_VECTOR_VT_R8> ] | [ <VT_VECTOR_VT_CY> ]
                    | [ <VT_VECTOR_VT_DATE> ] | [ <VT_VECTOR_VT_BSTR> ] | [
                    <VT_VECTOR_VT_ERROR> ] | [ <VT_VECTOR_VT_BOOL> ] | [
                    <VT_VECTOR_VT_VARIANT> ] | [ <VT_VECTOR_VT_I1> ] | [
                    <VT_VECTOR_VT_UI1> ] | [ <VT_VECTOR_VT_UI2> ] | [
                    <VT_VECTOR_VT_UI4> ] | [ <VT_VECTOR_VT_I8> ] | [ <VT_VECTOR_VT_UI8>
                    ] | [ <VT_VECTOR_VT_LPSTR> ] | [ <VT_VECTOR_VT_LPWSTR> ] | [
                    <VT_VECTOR_VT_FILETIME> ] | [ <VT_VECTOR_VT_CF> ] | [
                    <VT_VECTOR_VT_CLSID> ] | [ <VT_ARRAY_VT_I2> ] | [ <VT_ARRAY_VT_I4>
                    ] | [ <VT_ARRAY_VT_R4> ] | [ <VT_ARRAY_VT_R8> ] | [
                    <VT_ARRAY_VT_CY> ] | [ <VT_ARRAY_VT_DATE> ] | [ <VT_ARRAY_VT_BSTR>
                    ] | [ <VT_ARRAY_VT_ERROR> ] | [ <VT_ARRAY_VT_BOOL> ] | [
                    <VT_ARRAY_VT_VARIANT> ] | [ <VT_ARRAY_VT_DECIMAL> ] | [
                    <VT_ARRAY_VT_I1> ] | [ <VT_ARRAY_VT_UI1> ] | [ <VT_ARRAY_VT_UI2> ]
                    | [ <VT_ARRAY_VT_UI4> ] | [ <VT_ARRAY_VT_INT> ] | [
                    <VT_ARRAY_VT_UINT> ]

          <VT_EMPTY> ::= 0x0000

          <VT_NULL> ::= 0x0001

          <VT_I2> ::= 0x0002

          <VT_I4> ::= 0x0003

          <VT_R4> ::= 0x0004

          <VT_R8> ::= 0x0005

          <VT_CY> ::= 0x0006

          <VT_DATE> ::= 0x0007

          <VT_BSTR> ::= 0x0008

          <VT_ERROR> ::= 0x000A

          <VT_BOOL> ::= 0x000B

          <VT_DECIMAL> ::= 0x000E

          <VT_I1> ::= 0x0010

          <VT_UI1> ::= 0x0011

          <VT_UI2> ::= 0x0012

          <VT_UI4> ::= 0x0013

          <VT_I8> ::= 0x0014

          <VT_UI8> ::= 0x0015
```

```
<VT_INT> ::= 0x0016

<VT_UINT> ::= 0x0017

<VT_LPSTR> ::= 0x001E

<VT_LPWSTR> ::= 0x001F

<VT_FILETIME> ::= 0x0040

<VT_BLOB> ::= 0x0041

<VT_STREAM> ::= 0x0042

<VT_STORAGE> ::= 0x0043

<VT_STREAMED_Object> ::= 0x0044

<VT_STORED_Object> ::= 0x0045

<VT_BLOB_Object> ::= 0x0046

<VT_CF> ::= 0x0047

<VT_CLSID> ::= 0x0048

<VT_VERSIONED_STREAM> ::= 0x0049

<VT_VECTOR_VT_I2> ::= 0x1002

<VT_VECTOR_VT_I4> ::= 0x1003

<VT_VECTOR_VT_R4> ::= 0x1004

<VT_VECTOR_VT_R8> ::= 0x1005

<VT_VECTOR_VT_CY> ::= 0x1006

<VT_VECTOR_VT_DATE> ::= 0x1007

<VT_VECTOR_VT_BSTR> ::= 0x1008

<VT_VECTOR_VT_ERROR> ::= 0x100A

<VT_VECTOR_VT_BOOL> ::= 0x100B

<VT_VECTOR_VT_VARIANT> ::= 0x100C

<VT_VECTOR_VT_I1> ::= 0x1010

<VT_VECTOR_VT_UI1> ::= 0x1011

<VT_VECTOR_VT_UI2> ::= 0x1012
```

```
<VT_VECTOR_VT_UI4> ::= 0x1013

<VT_VECTOR_VT_I8> ::= 0x1014

<VT_VECTOR_VT_UI8> ::= 0x1015

<VT_VECTOR_VT_LPSTR> ::= 0x101E

<VT_VECTOR_VT_LPWSTR> ::= 0x101F

<VT_VECTOR_VT_FILETIME> ::= 0x1040

<VT_VECTOR_VT_CF> ::= 0x1047

<VT_VECTOR_VT_CLSID> ::= 0x1048

<VT_ARRAY_VT_I2> ::= 0x2002

<VT_ARRAY_VT_I4> ::= 0x2003

<VT_ARRAY_VT_R4> ::= 0x2004

<VT_ARRAY_VT_R8> ::= 0x2005

<VT_ARRAY_VT_CY> ::= 0x2006

<VT_ARRAY_VT_DATE> ::= 0x2007

<VT_ARRAY_VT_BSTR> ::= 0x2008

<VT_ARRAY_VT_ERROR> ::= 0x200A

<VT_ARRAY_VT_BOOL> ::= 0x200B

<VT_ARRAY_VT_VARIANT> ::= 0x200C

<VT_ARRAY_VT_DECIMAL> ::= 0x200E

<VT_ARRAY_VT_I1> ::= 0x2010

<VT_ARRAY_VT_UI1> ::= 0x2011

<VT_ARRAY_VT_UI2> ::= 0x2012

<VT_ARRAY_VT_UI4> ::= 0x2013

<VT_ARRAY_VT_INT> ::= 0x2016

<VT_ARRAY_VT_UINT> ::= 0x2017

<Padding> ::= “0” <!-- Must be set to zero -->

<Value> ::= <!-- Must be the value of the property represented and
```

```
                    serialized according to the value of Type -->

               <Dictionary> ::= *<DictionaryEntry>

               <DictionaryEntry> ::= <PropertyIdentifier> <Length> <Name>

               <Length> ::= <!-- Length of the Name field --> HEX_INTEGER

               <Name> ::= <!-- Value of the DictionaryEntry field -->  <Dictionary
                    Unicode Name> | <Dictionary Octet Name>

               <Dictionary Unicode Name ::= *<Unicode char> <NULL> <Unicode pad>

               <Unicode char> ::= UTF-16

               <Unicode pad> ::= 0+ <--! Padding such that <length> equals a multiple of
                    4>

               <NULL> ::= 0x00

               <Dictionary Octet Name> ::= *<char> <NULL>

               <char> ::= 8 bit byte
```

OFFICE 2003.4.9: END

OFFICE 2003.4.10: Footnotes and Endnotes

PRODUCT: WORD

GRAMMAR:

```
               <PlcfndRef> ::= <aCP2> <aFtnIdx>

               <aCP2> ::= <CP2> <aCP2>

               <CP2> ::= <Integer32> <!--Character Position: Specifies positions of
                    footnotes within the main document -->

               <aFtnIdx> ::= <Integer16> <aFtnIdx>
```

OFFICE 2003.4.10: END

OFFICE 2003.4.11: Headers and Footers

PRODUCT: WORD

GRAMMAR:

```
<Plcfhdd> ::= <aCP>

<aCP> ::= <CP> <aCP>

<CP> ::= <Integer32> <!--Character Position: Specifies the beginning of a
     story in the header document -->

<PlcfndTxt> ::= <aCP>
```

PRODUCT: EXCEL

GRAMMAR:

```
<Header> ::= <XLUnicodeString>

<HeaderFooter> ::= <FrtHeader> <GUID> <fHFDiffOddEven> <fHFDiffFirst>
     <fHFScaleWithDoc> <fHFAlignMargins> <unused> <cchHeaderEven>
     <cchFooterEven> <cchHeaderFirst> <cchFooterFirst> <strHeaderEven>
     <strFooterEven> <strHeaderFirst> <strFooterFirst>

<FrtHeader> ::= <rt> <FrtFlags> <reserved1>

<rt> ::= <0x089C>

<FrtFlags> ::= <fFrtRef> <fFrtAlert> <reserved2>

<fFrtRef> ::= <bit>

<fFrtAlert> ::= <bit>

<reserved2> ::= <bit-zero-ignored>[14]

<reserved1> ::= <octet-zero>[8]

<fHFDiffOddEven> ::= <bit>

<fHFDiffFirst> ::= <bit>

<fHFScaleWithDoc> ::= <bit>

<fHFAlignMargins> ::= <bit>

<unused> ::= <bit-ignored>[12]

<cchHeaderEven> ::= <Integer16>

<cchFooterEven> ::= <Integer16>

<cchHeaderFirst> ::= <Integer16>

<cchFooterFirst> ::= <Integer16>

<strHeaderEven> ::= <XLUnicodeString>
```

```
<strFooterEven> ::= <XLUnicodeString>

<strHeaderFirst> ::= <XLUnicodeString>

<strFooterFirst> ::= <XLUnicodeString>
```

PRODUCT: POWERPOINT

GRAMMAR:

```
<SlideHeadersFootersContainer> ::= <rh1> <HeadersFootersAtom>
     <UserDateAtom> <FooterAtom>

<rh1> ::= <0xF> <0x003> <RT_HeadersFooters>

<HeadersFootersAtom> ::= <rh2> <formatId> <fHasDate> <fHasTodayDate>
     <fHasUserDate> <fHasSlideNumber> <fHasHeader> <fHasFooter>
     <reserved1>

<rh2> ::= <0x0> <0x000> <RT_HeadersFootersAtom> <0x00000004>

<formatId> ::= <Integer16>

<fHasDate> ::= <bit>

<fHasTodayDate> ::= <bit>

<fUserDate> ::= <bit>

<fHasSlideNumber> ::= <bit>

<fHasHeader> ::= <bit>

<fHasFooter> ::= <bit>

<reserved1> ::= <bit-zero-ignored>[10]

<UserDateAtom> ::= <rh3> <PrintableUnicodeString>

<rh3> ::= <0x0> <0x000> <RT_CString> <rh.recLen> <!--rh.recLen must be an
     even number. It MUST be less than or equal to 510-->

<HeaderAtom> ::= <rh4> <PrintableUnicodeString>

<rh4> ::= <0x0> <0x001> <RT_CString> <rh.recLen> <!--rh.recLen must be an
     even number. -->

<FooterAtom> ::= <rh5> <PrintableUnicodeString>

<rh4> ::= <0x0> <0x002> <RT_CString> <rh.recLen> <!--rh.recLen must be an
     even number. -->

<NotesHeadersFootersContainer> ::= <rh6> <HeadersFootersAtom>
     <UserDateAtom> <HeaderAtom> <FooterAtom>
```

```
<rh6> ::= <0xF> <0x04> <RT_HeadersFooters>
```

OFFICE 2003.4.11: END

OFFICE 2003.4.12: Hidden Text (N/A)

OFFICE 2003.4.12: END

OFFICE 2003.4.13: Authors

PRODUCT: WORD

GRAMMAR:

```
<SttbfRMark> ::= <fExtend> <cData> <cbExtra> <Data>

<fExtend> ::= <0xFFFF>

<Data> ::= <cchData> <data> <extraData> | <cchData> <data> <extraData>
     <Data>

<cData> ::= <octet>[2] | <octet>[4]

<cbExtra> ::= <octet-zero>[2]

<cchData> ::= <octet>[2] <!-- based on fExtend -->

<data> ::= <octet>[2*cchData] | <octet>[cchData] <!--variable size -->

<extraData2> ::= <variable-empty>

<SttbAuthorAttrib> ::= <fExtend2> <cData2> <cbExtra2> <Data2>

<fExtend2> ::= <0xFFFF>

<Data2> ::= <cchData2> <data2> <extraData2> | <cchData2> <data2>
     <extraData2> <Data2>

<cData2> ::= <octet>[2]

<cbExtra2> ::= <0x0002>

<cchData2> ::= <octet>[2] <!-- based on fExtend2 -->

<data2> ::= <octet>[2*cchData] | <octet>[cchData] <!--variable size -->

<extraData2> ::= <variable-empty>
```

```
<SttbAuthorValue> ::= <fExtend3> <cData3> <cbExtra3> <Data3>

<fExtend3> ::= <0xFFFF>

<Data3> ::= <cchData3> <data3> <extraData3> | <cchData3> <data3>
     <extraData3> <Data3>

<cData3> ::= <octet>[2]

<cbExtra3> ::= <octet-zero>[2]

<cchData3> ::= <octet>[2] <!-- based on fExtend -->

<data3> ::= <octet>[2*cchData] | <octet>[cchData] <!--variable size -->

<extraData3> ::= <variable-empty>

<SttbfAssoc> ::= <fExtend4> <cData4> <cbExtra4> <Data4>

<fExtend4> ::= <0xFFFF>

<Data4> ::= <cchData4> <data4> <extraData4> | <cchData4> <data4>
     <extraData4> <Data4>

<cData4> ::= <0x0012>

<cbExtra4> ::= <octet-zero>[2]

<cchData4> ::= <octet>[2] <!-- based on fExtend -->

<data4> ::= <octet>[2*cchData] | <octet>[cchData] <!--variable size -->

<extraData4> ::= <variable-empty>

<PIDSI_AUTHOR> ::= <Type> <Padding> <Value>

<Type> ::= <0x001E> <!-- at offset 244 -->

<Padding> ::= <octet-zero>[2]

<Value> ::= <Size> <Characters>

<Size> ::= <0x000004>

<Characters> ::= <octet>[4]

<PIDSI_LASTAUTHOR> ::= <Type2> <Padding2> <Value2>

<Type2> ::= <0x001E> <!-- at offset 300 -->

<Padding2> ::= <octet-zero>[2]

<Value2> ::= <Size2> <Characters2>
```

```
<Size2> ::= <0x00000A>

<Characters2> ::= <octet>[12]
```

PRODUCT: EXCEL

GRAMMAR:

```
<UsrInfo> ::= <lUsrId> <GUID> <ShortDTR> <XLUnicodeString> <unused>

<1UsrId> ::= <Integer32>

<ShortDTR> ::= <Year> <Month> <Day> <Hour> <Minute> <Second> <Weekday>

<Year> ::= <Integer16>

<Month> ::= <Integer8>

<Day> ::= <Integer8>

<Hour> ::= <Integer8>

<Minute> ::= <Integer8>

<Second> ::= <Integer8>

<Weekday> ::= <Integer8>

<unused> ::= <octet-empty>

<Scenario> ::= <cref> <fLocked> <fHidden> <cchName> <cchComment>
     <cchNameUser> <rgchName> <rgchNameUser> <rgchComment> <rgSLC>
     <rgst> <unused2>

<cref> ::= <Integer16>

<fLocked> ::= <octet-boolean>

<fHidden> ::= <octet-boolean>

<cchName> ::= <Integer8>

<cchComment> ::= <Integer8>

<cchNameUser> ::= <Integer8>

<rgchName> ::= <XLUnicodeStringNoCch>

<rgchNameUser> ::= <XLUnicodeString> <!-- size <= 52 -->

<rgchNameComment> ::= <XLUnicodeString>

<rgSLC> ::= <SLCO8> | <SLCO8> <rgSLC>

<SLCO8> ::= <RwU> <ColSlco8U>
```

```
<RwU> ::= <octet>[2]

<ColSlco8U> ::= <octet>[2]

<rgst> ::= <XLUnicodeString> | <XLUnicodeString> <rgst>

<unused2> ::= <octet>[2*cref]
```

PRODUCT: POWEROINT

GRAMMAR:

```
<CurrentUserAtom> ::= <rh> <size> <headerToken> <offsetToCurrentEdit>
     <lenUserName> <docFileVersion> <majorVersion> <minorVersion>
     <unused> <ansiUserName> <relVersion> <unicodeUserName>

<rh> ::= <recVer> <recInstance> <RT_CurrentUserAtom>

<recVer> ::= <0x0>

<recInstance> ::= <0x000>

<size> ::= <0x00000014>

<headerToken> ::= <0xE391C05F> | <0xF3D1C4DF>

<offsetToCurrentEdit> ::= <octet>[4]

<lenUserName> ::= <Integer16> <!-- less than 256 -->

<docFileVersion> ::= <0x03F4>

<majorVersion> ::= <0x03>

<minorVersion> ::= <0x00>

<unused> ::= <octet-empty>[2]

<ansiUserName> ::= <PrintableAnsiString>

<relVersion> ::= <0x00000008> | <0x00000009>

<unicodeUserName> ::= <PrintableUnicodeString>

<BCUserNameAtom> ::= <rh2> <userName>

<rh2> ::= <0x0> <0x010> <RT_CString> <recLen> <!--rh.recLen must be an
     even number. It MUST be less than 510-->

<userName> ::= <FileOrDirNameFragment>
```

OFFICE 2003.4.13: END

OFFICE 2003.4.14: Routing Slip

PRODUCT: WORD

GRAMMAR:

```
<RouteSlipInfo> ::= <cbEntryID> <cbszName> <rgbEntryId> <szName>

<cbEntryID> ::= <Integer16> <!-- byte-size of rgbEntryId -->

<cbszName> ::= <Integer16> <!-- byte-size of szName -->

<rgbEntryId> ::= <octet>[cbEntryID]

<szName> ::= <octet>[cbszName]
```

PRODUCT: EXCEL

GRAMMAR:

```
<DocRoute> ::= <iStage> <cRecip> <delOption> <fRouted> <fReturnOrig>
     <fTrackStatus> <fCustomType> <unused1> <fSaveRouteInfo> <unused2>
     <cchSubject> <cchMessage> <cchRouteID> <cchCustType> <cchBookTitle>
     <cchOrg> <ulEIDSize> <szSubject> <szMessage> <szRouteID>
     <szCustType> <szBookTitle> <szOrg> <rgchSSAddr>

<iStage> ::= <Integer16> <!-- <= (cRecip+1) -->

<cRecip> ::= <Integer16> <!-- count of RecipName following this record --
     >

<delOption> ::= <octet-boolean>[2]

<fRouted> ::= <bit>

<fReturnOrig> ::= <bit>

<fTrackStatus> ::= <bit>

<fCustomType> ::= <bit>

<unused1> ::= <bit-ignored>[3]

<fSaveRouteInfo> ::= <bit>

<unused2> ::= <bit-ignored>[8]

<cchSubject> ::= <Integer16> <!-- < 257 -->

<cchMessage> ::= <Integer16> <!-- < 257 -->

<cchRouteID> ::= <Integer16> <!-- < 257 -->

<cchCustType> ::= <Integer16> <!-- < 257 -->
```

```
<cchBookTitle> ::= <Integer16> <!-- < 257 -->

<cchOrg> ::= <Integer16> <!-- < 257 -->

<ulEIDSize> ::= <Integer32> <!-- < 8203 -->

<RecipName> ::= <cchRecip> <ulEIDSize> <szFriendly> <rgchSSAddr>

<cchRecip> ::= <Integer16> <!-- < 257 byte-size of szFriendly -->

<ulEIDSize> ::= <Integer32> <!-- byte-size of rgchSSAddr -->

<szFriendly> ::= <octet>[cchRecip]

<rgchSSAddr> ::= <octet>[ulEIDSize]
```

PRODUCT: POWERPOINT

GRAMMAR:

```
<DocRoutingSlipAtom> ::= <rh> <length> <unused1> <recipientCount>
     <currentRecipient> <fOneAfterAnother> <rReturnWhenDone>
     <fTrackStatus> <reserved1> <fDocumentRouted> <fCycleCompleted>
     <reserved2> <unused2> <originatorString>
     <rgRecipientRoutingSlipStrings> <subjectString> <messageString>
     <unused3>

<rh> ::= <0x0> <0x000> <RT_DocRoutingSlipAtom>

<length> ::= <Integer32>

<unused1> ::= <octet-ignored>[4]

<recipientCount> ::= <Integer32> <!-- count of strings in
     rgRecipientRoutingSlipStrings -->

<currentRecipient> ::= <Integer32> <!-- index of the addressee, <=
     (recipientCount+1) -->

<fOneAfterAnother> ::= <bit>

<fReturnWhenDone> ::= <bit>

<fTrackStatus> ::= <bit>

<reserved1> ::= <bit-zero-ignored>

<fDocumentRouted> ::= <bit>

<fCycleCompleted> ::= <bit>

<reserved2> ::= <bit-zero-ignored>[26]

<unused2> ::= <octet-ignored>[4]
```

```
<originatorString> ::= <DocRoutingSlipString>

<rgRecipientRoutingSlipStings> ::= <DocRoutingSlipString> |
     <DocRoutingSlipString> <rgRecipientRoutingSlipStings>

<DocRoutingSlipString> ::= <stringType> <stringLength> <string>

<stringType> ::= <0x0001> | <0x0002> | <0x0003> | <0x0004>

<stringLength> ::= <Integer16> <!-- byte size of (string-1) -->

<string> ::= <PrintableAnsiString>
```

OFFICE 2003.4.14: END

OFFICE 2003.4.15: Printer Information

PRODUCT: WORD

GRAMMAR:

```
<PrDrvr> ::= <szPrinter> <szPrPort> <szPrDriver> <szTruePrnName>

<szPrinter> ::= <octet-null-terminated> <!-- printer used by the computer
     or network -->

<szPrPort> ::= <octet-null-terminated> <!-- string of the printer port --
     >

<szPrDriver> ::= <octet-null-terminated> <!-- string specifying the
     printer driver -->

<szTruePrnName> ::= <octet-null-terminated> <!-- string of the printer
     name from the manufacturer -->
```

PRODUCT: EXCEL

GRAMMAR:

```
<Pls> ::= <reserved> <rgb>

<reserved> ::= <octet-zero>[2]

<rgb> ::= <DEVMODE>

<DEVMODE> ::= <dmDeviceName> <dmSpecVersion> <dmDriverVersion> <dmSize>
     <dmDriverExtra> <dmFields> <dmOrientation> <dmPaperSize>
     <dmPaperLength> <dmPaperWidth> <dmScale> <dmCopies>
     <dmDefaultSource> <dmPrintQuality> <dmPosition>
     <dmDisplayOrientation> <dmDisplayFixedOutput> <dmColor> <dmDuplex>
```

```
                    <dmYResolution> <dmTTOption> <dmCollate> <dmFormName> <dmLogPixels>
                    <dmBitsPerPel> <dmPelsWidth> <dmPelsHeight> <dmDisplayFlags>
                    <dmNup> <dmDisplayFrequency> <dmICMMethod> <dmICMIntent>
                    <dmMediaType> <dmDitherType> <dmReserved1> <dmReserved2>
                    <dmPanningWidth> <dmPanningHeight>
```

OFFICE 2003.4.15: END

OFFICE 2003.4.16: Smart Tags

COMMON SMART TAGS BNF PRODUCTIONS

GRAMMAR:

```
<PropertyBagStore> ::= <cFactoidType> <factoidTypes> <cbHdr> <sVer>
     <cFactoid> <cste> <stringTable>

<cFactoidType> ::= <Integer32> <!-- size of factoidTypes -->

<factoidTypes> ::= <FactoidType> | <FactoidType> <factoidTypes>

<FactoidType> ::= <cbFactoid> <PropertyBagId> <rgbUri> <rgbTag>
     <rgbDownLoadURL>

<cbFactoid> ::= <Integer32> <!-- size (in bytes) of FactoidType minus
     itself -->

<rgbUri> ::= <PBString>

<rgbTag> ::= <PBString>

<rgbDownLoadURL> ::= <PBString>

<PropertyBag> ::= <PropertyBagId> <cProp> <cbUnknown> <properties>

<PropertyBagId> ::= <Integer16>

<cProp> ::=  <Integer16> <!-- size of properties -->

<cbUnknown> ::= <octet-zero-ignored>

<properties> ::= <Property> | <Property> <properties>

<Property> ::= <keyIndex> <valueIndex>

<keyIndex> ::= <Integer32>

<valueIndex> ::= <Integer32>

<PBString> ::= <cch> <fAnsiString> <rgxch>
```

```
<cch> ::= <bit>[15] <!-- size of rgxch -->

<fAnsiString> ::= <bit>

<rgxch> ::= <octet-null-terminated>
```

PRODUCT: WORD

GRAMMAR:

```
<SmartTagData> ::= <PropertyBagStore> <propBags>

<propBags> ::= <PropertyBag> | <PropertyBag> <propBags>
```

PRODUCT: EXCEL

GRAMMAR:

In Excel, Feat structures store data regarding shared features. A type of Feat is FeatSmartTag, which contains Smart Tag information.

```
<Feat> <!-- for Smart Tags --> ::= <FrtHeader> <ISFFACTOID> <!-- a
     SharedFeatureType enumeration--> <reserved1> <reserved2> <cref>
     <cbFeatData> <reserved3> <refs> <FeatSmartTag>

<FrtHeader> <!-- for Feat --> ::= <0x0868> <FrtFlags> <reserved4>

<FrtFlags> ::= <fFrtRef> <fFrtAlert> <reserved5>

<fFrtRef> ::= <bit>

<fFrtAlert> ::= <bit>

<reserved5> ::= <bit-zero-ignored>

<reserved4> ::= <octet-zero-ignored>

<FeatSmartTag> ::= <hashValue> <cSmartTags> <rgFactoid>

<hashValue> ::= <Integer32>

<cSmartTags> ::= <Integer8> <!-- number of rgFactoids -->

<rgFactoid> ::= <FactoidData> | <FactoidData> <rgFactoid>

<FactoidData> ::= <fDelete> <fXMLBased> <reserved6> <PropertyBag>

<fDelete> ::= <bit>

<fXMLBased> ::= <bit>

<reserved6> ::= <bit-zero-ignored>
```

PRODUCT: POWERPOWER POINT

GRAMMAR:

```
<PP11DocBinaryTagExtension> ::= <rh1> <___PPT11> <rhData>
```

```
                    <SmartTagStore11Container> <OutlineTextProps11Container>

               <rh1> ::= <0x0> <0x000> <RT_CString> <0x00000010>

               <rhData> ::= <0x0> <0x000> <RT_BinaryTagDataBlob>

               <SmartTagStore11Container> ::= <rh2> <cBags> <PropertyBagStore>
                    <rgPpropBag>

               <rh2> ::= <0xF> <0x000> <RT_SmartTagStore11Container>

               <cBags> ::= <Integer32> <!-- size of rgPpropBag -->

               <rgPpropBag> ::= <PropertyBag> | <PropertyBag> <rgPpropBag>

               <OutlineTextProps11Container> ::= <!-- specifies text properties -->
```

OFFICE 2003.4.16: END

OFFICE 2003.4.17: Meeting Minder (N/A)

OFFICE 2003.4.17: END

OFFICE 2003.4.18: Image Properties

PRODUCT: WORD, EXCEL, POWERPOINT

GRAMMAR:

```
               <OfficeArtDggContainer> :: = <OfficeArtRecordHeader> <OfficeArtFDGGBlock>
                    <OfficeArtBStoreContainer> <OfficeArtFOPT> <OfficeArtTertiaryFOPT>
                    <OfficeArtColorMRUContainer> <OfficeArtSplitMenuColorContainer>

               <OfficeArtSpContainer> ::= <OfficeArtRecordHeader> <OfficeArtFSPGR>
                    <OfficeArtFSP> <OfficeArtFPSPL> <OfficeArtFOPT>
                    <OfficeArtSecondaryFOPT> <OfficeArtTertiaryFOTP>
                    <OfficeArtChildAnchor> <OfficeArtClientAnchor>
                    <OfficeArtClientData> <OfficeArtClientTextbox>
                    <OfficeArtSecondaryFOPT> <OfficeArtTertiaryFOTP>

               <OfficeArtFOPT> ::= <OfficeArtRecordHeader> <OfficeArtRGFOPTE>

               <OfficeArtSecondaryFOPT> ::= <OfficeArtRecordHeader> <OfficeArtRGFOPTE>

               <OfficeArtTertiaryFOPT> ::= <OfficeArtRecordHeader> <OfficeArtRGFOPTE>

               <OfficeArtRGFOPTE> ::= *<OfficeArtFOPTE> <complexData> <!-- The
                    complexData record holds additional property values that cannot be
                    contained within the OfficeArtFOPTE's op record. -->
```

```
<OfficeArtFOPTE> ::= <OfficeArtFOPTEOPID> <op> <!-- This key-value pair
     represents a predefined property, OfficeArtFOPTEOPID, and the
     corresponding value. See [MS-ODRAW].pdf Section 2.2.7 for more
     details.-->

<OfficeArtFOPTEOPID> :: = <3D-Object Boolean Properties> | <3D-Style
     Boolean Properties> | <adjust2Value> | <adjust3Value> |
     <adjust4Value> | <adjust5Value> | <adjust6Value> | <adjust7Value> |
     <adjust8Value> | <adjustValue> | <alignHR> | <anchorText> | <Blip
     Boolean Properties> | <borderBottomColor> | <borderLeftColor> |
     <borderRightColor> | <borderTopColor> | <bottom> | <Bottom Line
     Style Boolean Properties> | <bWMode> | <bWModeBW> | <bWModePureBW>
     | <c3DAmbientIntensity> | <c3DCrMod> | <c3DDiffuseAmt> |
     <c3DEdgeThickness> | <c3DExtrudeBackward> | <c3DExtrudeForward> |
     <c3DExtrusionColor> | <c3DExtrusionColorExt> |
     <c3DExtrusionColorExtMod> | <c3DFillIntensity> | <c3DFillX> |
     <c3DFillY> | <c3DFillZ> | <c3DKeyIntensity> | <c3DKeyX> | <c3DKeyY>
     | <c3DKeyZ> | <c3DOriginX> | <c3DOriginY> | <c3DRenderMode> |
     <c3DRotationAngle> | <c3DRotationAxisX> | <c3DRotationAxisY> |
     <c3DRotationAxisZ> | <c3DRotationCenterX> | <c3DRotationCenterY> |
     <c3DRotationCenterZ> | <c3DShininess> | <c3DSkewAmount> |
     <c3DSkewAngle> | <c3DSpecularAmt> | <c3DTolerance> |
     <c3DXRotationAngle> | <c3DXViewpoint> | <c3DYRotationAngle> |
     <c3DYViewpoint> | <c3DZViewpoint> | <Callout Boolean Properties> |
     <cdirFont> | <cropFromBottom> | <cropFromLeft> | <cropFromRight> |
     <cropFromTop> | <cxk> | <cxstyle> | <dgmBaseTextScale> |
     <dgmConstrainBounds> | <dgmDefaultFontSize> | <dgmLayout> |
     <dgmLayoutMRU> | <dgmNodeKind> | <dgmScaleX> | <dgmScaleY> |
     <dgmStyle> | <dgmt> | <dhgt> | <Diagram Boolean Properties> |
     <dxHeightHR> | <dxTextLeft> | <dxTextRight> | <dxWidthHR> |
     <dxWrapDistLeft> | <dxWrapDistRight> | <dxyCalloutDropSpecified> |
     <dxyCalloutGap> | <dxyCalloutLengthSpecified> | <dyTextBottom> |
     <dyTextTop> | <dyWrapDistBottom> | <dyWrapDistTop> | <equationXML>
     | <Fill Style Boolean Properties> | <fillAngle> | <fillBackColor> |
     <fillBackColorExt> | <fillBackColorExtMod> | <fillBackOpacity> |
     <fillBlip> | <fillBlipFlags> | <fillBlipName> | <fillColor> |
     <fillColorExt> | <fillColorExtMod> | <fillCrMod> | <fillDztype> |
     <fillFocus> | <fillHeight> | <fillOpacity> | <fillOriginX> |
     <fillOriginY> | <fillRectBottom> | <fillRectLeft> | <fillRectRight>
     | <fillRectTop> | <fillShadeColors> | <fillShadePreset> |
     <fillShadeType> | <fillShapeOriginX> | <fillShapeOriginY> |
     <fillToBottom> | <fillToLeft> | <fillToRight> | <fillToTop> |
     <fillType> | <fillWidth> | <geoBottom> | <geoLeft> | <Geometry
     Boolean Properties> | <Geometry Text Boolean Properties> |
     <geoRight> | <geoTop> | <Group Shape Boolean Properties> |
     <gtextAlign> | <gtextCSSFont> | <gtextFont> | <gtextSize> |
     <gtextSpacing> | <gtextUNICODE> | <gvPage> | <gvRelPage> |
     <hspMaster> | <hspNext> | <idDiscussAnchor> | <Ink Boolean
     Properties> | <left> | <Left Line Style Boolean Properties> |
     <lidRegroup> | <Line Style Boolean Properties> | <lineBackColor> |
     <lineBackColorExt> | <lineBackColorExtMod> | <lineBottomBackColor>
     | <lineBottomBackColorExt> | <lineBottomBackColorExtMod> |
     <lineBottomColor> | <lineBottomColorExt> | <lineBottomColorExtMod>
     | <lineBottomCrMod> | <lineBottomDashing> | <lineBottomDashStyle> |
     <lineBottomEndArrowhead> | <lineBottomEndArrowLength> |
     <lineBottomEndArrowWidth> | <lineBottomEndCapStyle> |
     <lineBottomFillBlip> | <lineBottomFillBlipFlags> |
     <lineBottomFillBlipName> | <lineBottomFillDztype> |
     <lineBottomFillHeight> | <lineBottomFillWidth> |
```

<lineBottomJoinStyle> | <lineBottomMiterLimit> |
<lineBottomOpacity> | <lineBottomStartArrowhead> |
<lineBottomStartArrowLength> | <lineBottomStartArrowWidth> |
<lineBottomStyle> | <lineBottomType> | <lineBottomWidth> |
<lineColor> | <lineColorExt> | <lineColorExtMod> | <lineCrMod> |
<lineDashing> | <lineDashStyle> | <lineEndArrowhead> |
<lineEndArrowLength> | <lineEndArrowWidth> | <lineEndCapStyle> |
<lineFillBlip> | <lineFillBlipFlags> | <lineFillBlipName> |
<lineFillDztype> | <lineFillHeight> | <lineFillWidth> |
<lineJoinStyle> | <lineLeftBackColor> | <lineLeftBackColorExt> |
<lineLeftBackColorExtMod> | <lineLeftColor> | <lineLeftColorExt> |
<lineLeftColorExtMod> | <lineLeftCrMod> | <lineLeftDashing> |
<lineLeftDashStyle> | <lineLeftEndArrowhead> |
<lineLeftEndArrowLength> | <lineLeftEndArrowWidth> |
<lineLeftEndCapStyle> | <lineLeftFillBlip> |
<lineLeftFillBlipFlags> | <lineLeftFillBlipName> |
<lineLeftFillDztype> | <lineLeftFillHeight> | <lineLeftFillWidth> |
<lineLeftJoinStyle> | <lineLeftMiterLimit> | <lineLeftOpacity> |
<lineLeftStartArrowhead> | <lineLeftStartArrowLength> |
<lineLeftStartArrowWidth> | <lineLeftStyle> | <lineLeftType> |
<lineLeftWidth> | <lineMiterLimit> | <lineOpacity> |
<lineRightBackColor> | <lineRightBackColorExt> |
<lineRightBackColorExtMod> | <lineRightColor> | <lineRightColorExt>
| <lineRightColorExtMod> | <lineRightCrMod> | <lineRightDashing> |
<lineRightDashStyle> | <lineRightEndArrowhead> |
<lineRightEndArrowLength> | <lineRightEndArrowWidth> |
<lineRightEndCapStyle> | <lineRightFillBlip> |
<lineRightFillBlipFlags> | <lineRightFillBlipName> |
<lineRightFillDztype> | <lineRightFillHeight> |
<lineRightFillWidth> | <lineRightJoinStyle> | <lineRightMiterLimit>
| <lineRightOpacity> | <lineRightStartArrowhead> |
<lineRightStartArrowLength> | <lineRightStartArrowWidth> |
<lineRightStyle> | <lineRightType> | <lineRightWidth> |
<lineStartArrowhead> | <lineStartArrowLength> |
<lineStartArrowWidth> | <lineStyle> | <lineTopBackColor> |
<lineTopBackColorExt> | <lineTopBackColorExtMod> | <lineTopColor> |
<lineTopColorExt> | <lineTopColorExtMod> | <lineTopCrMod> |
<lineTopDashing> | <lineTopDashStyle> | <lineTopEndArrowhead> |
<lineTopEndArrowLength> | <lineTopEndArrowWidth> |
<lineTopEndCapStyle> | <lineTopFillBlip> | <lineTopFillBlipFlags> |
<lineTopFillBlipName> | <lineTopFillDztype> | <lineTopFillHeight> |
<lineTopFillWidth> | <lineTopJoinStyle> | <lineTopMiterLimit> |
<lineTopOpacity> | <lineTopStartArrowhead> |
<lineTopStartArrowLength> | <lineTopStartArrowWidth> |
<lineTopStyle> | <lineTopType> | <lineTopWidth> | <lineType> |
<lineWidth> | <lTxid> | <metroBlob> | <movie> | <OfficeArtFOPTE> |
<pAdjustHandles> | <pConnectionSites> | <pConnectionSitesDir> |
<pctHoriz> | <pctHorizPos> | <pctHR> | <pctVert> | <pctVertPos> |
<Perspective Style Boolean Properties> | <perspectiveOffsetX> |
<perspectiveOffsetY> | <perspectiveOriginX> | <perspectiveOriginY>
| <perspectivePerspectiveX> | <perspectivePerspectiveY> |
<perspectiveScaleXToX> | <perspectiveScaleXToY> |
<perspectiveScaleYToX> | <perspectiveScaleYToY> | <perspectiveType>
| <perspectiveWeight> | <pGuides> | <pib> | <pibFlags> | <pibName>
| <pibPrint> | <pibPrintFlags> | <pibPrintName> |
<pictureBrightness> | <pictureContrast> | <pictureDblCrMod> |
<pictureFillCrMod> | <pictureId> | <pictureLineCrMod> |
<pictureRecolor> | <pictureRecolorExt> | <pictureRecolorExtMod> |
<pictureTransparent> | <pictureTransparentExt> |

<pictureTransparentExtMod> | <pihlShape> | <pInkData> | <pInscribe> | <posh> | <posrelh> | <posrelv> | <posv> | <pRelationTbl> | <Protection Boolean Properties> | <pSegmentInfo> | <pVertices> | <pWrapPolygonVertices> | <Relative Transform Boolean Properties> | <relBottom> | <relLeft> | <relRight> | <relRotation> | <relTop> | <reserved1370> | <reserved1372> | <reserved1374> | <reserved1376> | <reserved1377> | <reserved1378> | <reserved1434> | <reserved1436> | <reserved1438> | <reserved1440> | <reserved1441> | <reserved1442> | <reserved1498> | <reserved1500> | <reserved1502> | <reserved1504> | <reserved1505> | <reserved1506> | <reserved1562> | <reserved1564> | <reserved1566> | <reserved1568> | <reserved1569> | <reserved1570> | <reserved278> | <reserved280> | <reserved281> | <reserved284> | <reserved286> | <reserved287> | <reserved415> | <reserved417> | <reserved419> | <reserved421> | <reserved422> | <reserved423> | <reserved474> | <reserved476> | <reserved478> | <reserved480> | <reserved481> | <reserved482> | <reserved531> | <reserved533> | <reserved535> | <reserved537> | <reserved538> | <reserved539> | <reserved646> | <reserved650> | <reserved652> | <reserved653> | <right> | <Right Line Style Boolean Properties> | <rotation> | <scriptLang> | <Shadow Style Boolean Properties> | <shadowColor> | <shadowColorExt> | <shadowColorExtMod> | <shadowCrMod> | <shadowHighlight> | <shadowHighlightExt> | <shadowHighlightExtMod> | <shadowOffsetX> | <shadowOffsetY> | <shadowOpacity> | <shadowOriginX> | <shadowOriginY> | <shadowSecondOffsetX> | <shadowSecondOffsetY> | <shadowSoftness> | <shadowType> | <Shape Boolean Properties> | <shapePath> | <Signature Line Boolean Properties> | <sizerelh> | <sizerelv> | <spcoa> | <spcod> | <tableProperties> | <tableRowProperties> | <Text Boolean Properties> | <top> | <Top Line Style Boolean Properties> | <Transform Boolean Properties> | <txdir> | <txflTextFlow> | <Unknown HTML Boolean Properties> | <unused134> | <unused140> | <unused141> | <unused832> | <unused906> | <Web Component Boolean Properties> | <webComponentWzHtml> | <webComponentWzName> | <webComponentWzUrl> | <WrapText> | <wzCalloutId> | <wzDescription> | <wzFillId> | <wzFormulaeId> | <wzGtextId> | <wzHandlesId> | <wzLineId> | <wzLockId> | <wzName> | <wzPathId> | <wzPerspectiveId> | <wzPictureId> | <wzScript> | <wzScriptExtAttr> | <wzScriptLangAttr> | <wzShadowId> | <wzSigSetupAddlXml> | <wzSigSetupId> | <wzSigSetupProvId> | <wzSigSetupProvUrl> | <wzSigSetupSignInst> | <wzSigSetupSuggSigner> | <wzSigSetupSuggSigner2> | <wzSigSetupSuggSignerEmail> | <wzTextId> | <wzThreeDId> | <wzTooltip> | <wzWebBot> | <xLimo> | <yLimo>

PRODUCT: WORD

GRAMMAR:

<FibRgFcLcb97> ::= … <fcDggInfo> <lcbDggInfo> … (See [MS-DOC].pdf Section 2.5.6)

<OfficeArtContent> ::= <OfficeArtDggContainer> *<OfficeArtWordDrawing>

<OfficeArtWordDrawing> ::= <dgglbl> <OfficeArtDgContainer>

<OfficeArtDgContainer> ::= <OfficeArtRecordHeader> <OfficeArtFDG> <OfficeArtFRITContainer> <OfficeArtSpgrContainer> <OfficeArtSpContainer> <OfficeArtSpgrContainerFileBlock> <OfficeArtSolverContainer>

```
<PlcfSpa> ::= *<CP> <Spa>

<Spa> ::= <lid> <rca> <fHdr> <bx> <by> <wr> <wrk> <fRcaSimple>
     <fBelowText> <fAnchorLock> <cTxbx>

<lid> ::= <!-- An integer that specifies an id of a shape in the
     OfficeArtDggContainer structure. See [MS-DOC].pdf Section 2.9.250
     for details. -->

<rca>  ::= <!-- An Rca structure that specifies the rectangle where the
     drawing exists. See [MS-DOC].pdf Section 2.9.219 for details. -->

<fHdr> ::= <!-- Undefined and ignored bit. -->

<bx> ::= <!-- An unsigned integer that specifies the horizontal position
     of the origin used to calculate the rca.-->

<by> ::= <!-- An unsigned integer that specifies the vertical position of
     the origin used to calculate the rca.-->

<wr> ::= <!-- An unsigned integer that specifies the style of text
     wrapping around this shape. -->

<wrk> ::= <!-- An unsigned integer that specifies the details of the text
     wrapping around this shape. -->

<fRcaSimple> ::= <!-- Undefined and ignored bit. -->

<fBelowText> ::= <!-- An unsigned integer that specifies whether this
     shape is behind the text. A value of 1 specifies that the shape
     will appear behind the paragraph. A value of 0 specifies that the
     shape will appear in front of the text and obscure it. If wr is not
     3, this field MUST be ignored. -->

<fAnchorLock> ::= <!-- An unsigned integer that specifies whether the
     shape‘s anchor is locked to its current paragraph. -->

<cTxbx>  ::= <!-- Undefined and ignored bit. -->
```

PRODUCT: EXCEL

GRAMMAR:

```
<WORKSHEET> ::= <BOF> <WORKSHEETCONTENT>

<WORKSHEETCONTENT> ::= [<Uncalced>] <Index> <GLOBALS> <PAGESETUP>
     [<HeaderFooter>] [<BACKGROUND>] *<BIGNAME> [<PROTECTION>] <COLUMNS>
     [<SCENARIOS>] <SORTANDFILTER> <Dimensions> [<CELLTABLE>] <OBJECTS>
     *<HFPicture> *<Note> *<PIVOTVIEW> [<DCON>] 1*<WINDOW> *<CUSTOMVIEW>
     *2<SORT> [<DxGCol>] *<MergeCells> [<LRng>] *<QUERYTABLE>
     [<PHONETICINFO>] <CONDFMTS> *<HLINK> [<DVAL>] [<CodeName>]
     *<WebPub> *<CellWatch> [<SheetExt>] *<FEAT> *<FEAT11> *<RECORD12>
     <EOF>

<CHARTSHEET> ::= <BOF> <CHARTSHEETCONTENT>

<CHARTSHEETCONTENT> ::= [<WriteProtect>] [<SheetExt>] [<WebPub>]
```

```
            *<HFPicture> <PAGESETUP> <PrintSize> [<HeaderFooter>]
            [<BACKGROUND>] *<Fbi> *<Fbi2> [<ClrtClient>] [<PROTECTION>]
            [<Palette>] [<SXViewLink>] [<PivotChartBits>] [<SBaseRef>]
            [<MsoDrawingGroup>] <OBJECTS> <Units> <CHARTFOMATS> <SERIESDATA>
            *<WINDOW> *<CUSTOMVIEW> [<CodeName>] [<CRTMLFRT>] <EOF>

      <DIALOGSHEET> ::= <BOF> <DIALOGSHEETCONTENT>

      <DIALOGSHEETCONTENT> ::= [<Uncalced>] <Index> <GLOBALS> <PAGESETUP>
            [<HeaderFooter>] *<BIGNAME> [<DIALOGPROTECTION>] <DefColWidth>
            <Dimensions> <OBJECTS> *<HFPicture> *<Note> 1*<DIALOGWINDOW>
            *<DIALOGCUSTOMVIEW> [<CodeName>] [<SheetExt>] *<RECORD12> <EOF>

      <WORKBOOK> ::= <BOF> <WORKBOOKCONTENT>

      <WORKBOOKCONTENT> ::= [<WriteProtect>] [<FilePass>] [<Template>]
            <INTERFACE> <WriteAccess> [<FileSharing>] <CodePage> *2047Lel DSF
            [Excel9File] RRTabId [ObProj] [ObNoMacros] [CodeName] [FNGROUPS]
            *Lbl [OleObjectSize] PROTECTION 1*Window1 Backup HideObj Date1904
            CalcPrecision RefreshAll BookBool FORMATTING
            *(PIVOTCACHEDEFINITION) [DOCROUTE] *UserBView UsesELFs
            1*BUNDLESHEET METADATA [MTRSettings] [ForceFullCalculation] Country
            *SUPBOOK *LBL *RTD [RecalcId] *HFPicture *MSODRAWINGGROUP
            [SHAREDSTRINGS] ExtSST *WebPub [WOpt] [CrErr] [BookExt] *FeatHdr
            *DConn [THEME] [CompressPictures] [Compat12] [GUIDTypeLib] EOF

      <WORKBOOKCONTENT> ::= [<WriteProtect>] [<FilePass>] [<Template>]
            <INTERFACE> <WriteAccess> [<FileSharing>] <CodePage> *2047<Lel>
            <DSF> [<Excel9File>] <RRTabId> [<ObProj>] [<ObNoMacros>]
            [<CodeName>] [<FNGROUPS>] *<Lbl> [<OleObjectSize>] [<PROTECTION>]
            1*<Window1> <Backup> <HideObj> <DateObj> <Date1904> <CalcPrecision>
            <RefreshAll> <BookBool> <FORMATTING> *<PIVOTCACHEDEFINITION>
            [<DOCROUTE>] *<UserBView> <UsesELFs> 1*<BUNDLESHEET> <METADATA>
            [<MTRSettings>] [<ForceFullCalculation>] <Country> *<SUPBOOK>
            *<LBL> *<RTD> [<RecalcId>] *<HFPicture> *<MSODRAWINGGROUP>
            [<SHAREDSTRINGS>] <ExtSST> *<WebPub> [<WOpt>] [<CrErr>] [<BookExt>]
            *<FeatHdr> *<DConn> [<THEME>] [<CompressPictures>] [<Compat12>]
            [<GUIDTypeLib>] <EOF>

      <MACROSHEET> ::= <BOF> <MACROSHEETCONTENT>

      <MACROSHEETCONTENT> ::= [<Uncalced>] <Index> [<Intl>] <GLOBALS>
            <PAGESETUP> [<HeaderFooter>] [<BACKGROUND>] *<BIGNAME>
            [<PROTECTION>] <COLUMNS> <MACROSORTANDFILTER> <Dimensions>
            [<CELLTABLE>] <OBJECTS> *<HFPicture> *<Note> [<DCON>] 1*<WINDOW>
            *<CUSTOMVIEW> *2<SORT> [<DxGCol>] [<PHONETICINFO>] [<CodeName>]
            *<CellWatch> [<SheetExt>] *<FEAT> *<RECORD12> <EOF>

      <HFPicture> ::= <FrtHeader> <unused> <reserved> [<OfficeArtDggContainer>
            | <OfficeArtDgContainer> ]

      <MSODRAWINGGROUP> ::= <MsoDrawingGroup> *<Container>

      <MsoDrawingGroup> ::= <OfficeArtDggContainer>
```

PRODUCT: POWERPOINT

GRAMMAR:

```
<DocumentContainer> ::= RecordHeader> <DocumentAtom> <ExObjListContainer>
     <DocumentTextInfoContainer> <SoundCollectionContainer>
     <DrawingGroupContainer> <MasterListWithTextContainer>
     <DocInfoListContainer> <SlideHeadersFootersContainer>
     <NotesHeadersFootersContainer> <SlideListWithTextContainer>
     <NotesListWithTextContainer> <SlideShowDocInfoAtom>
     <NamedShowsContainer> <SummaryContainer> <DocRoutingSlipAtom>
     <PrintOptionsAtom> <RoundTripCustomTableStyles12Atom>
     <EndDocumentAtom> <rtCustomTableStylesAtom2>

<DrawingGroupContainer> ::= <RecordHeader> <OfficeArtDggContainer>

<HandoutContainer> ::= <RecordHeader> <DrawingContainer>
     <SlideSchemeColorSchemeAtom> <SlideNameAtom>
     <SlideProgTagsContainer> <HandoutRoundTripAtom>

<MainMasterContainer> ::= <RecordHeader> <SlideAtom>
     *<SchemeListElementColorSchemeAtom> <TextMasterStyleAtom>
     <RoundTripOArtTextStyles12Atom> <SlideShowSlideInfoAtom>
     <PerSlideHeadersFootersContainer> <DrawingContainer>
     <SlideSchemeColorSchemeAtom> <SlideNameAtom>
     <SlideProgTagsContainer> <RoundTripMainMasterRecord>
     <TemplateNameAtom>

<NotesContainer> ::= <RecordHeader> <NotesAtom> <DrawingContainer>
     <SlideSchemeColorSchemeAtom> <SlideNameAtom>
     <SlideProgTagsContainer> <NotesRoundTripAtom>

<SlideContainer> ::= <SlideAtom> <SlideShowSlideInfoAtom>
     <PerSlideHeadersFootersContainer>
     <RoundTripSlideSyncInfo12Container> <DrawingContainer>
     <SlideSchemeColorSchemeAtom> <SlideNameAtom>
     <SlideProgTagsContainer> <RoundTripSlideRecord>

<DrawingContainer> ::= <RecordHeader> <OfficeArtDgContainer>
```

OFFICE 2003.4.18: END

OFFICE 2003.4.19: Windows Metafile (WMF) and Enhanced Metafile (EMF)

PRODUCT: WORD, EXCEL, POWERPOINT

GRAMMAR:

```
<OfficeArtDggContainer> ::= <OfficeArtRecordHeader1> <OfficeArtFDGGBlock>
     <OfficeArtBStoreContainer> <OfficeArtFOPT> <OfficeArtTertiaryFOPT>
     <OfficeArtColorMRUContainer> <OfficeArtSplitMenuColorContainer>

<OfficeArtRecordHeader> ::= <OfficeArtRecordHeader1> |
     <OfficeArtRecordHeader2> | <OfficeArtRecordHeader3> |
```

```
        <OfficeArtRecordHeader4>

    <OfficeArtRecordHeader1> ::= <recVer1> <recInstance1> <recType1>
        <recLen1>

    <recVer1> ::= 0xF

    <recInstance1> ::= 0x000

    <recType1> ::= 0xF000

    <recLen1> ::= <!-- An unsigned integer that specifies the number of bytes
        following the header which contains document-wide file records -->

    <OfficeArtBStoreContainer> ::= <OfficeArtRecordHeader2>
        <OfficeArtBStoreContainerFileBlock>+

    <OfficeArtRecordHeader2> ::= <recVer2> <recInstance2> <recType2>
        <recLen2>

    <recVer2> ::= 0xF

    <recInstance2> ::= <!-- An unsigned integer that specifies the number of
        contained OfficeArtBStoreContainerFileBlock records. -->

    <recType2> ::= 0xF001

    <recLen2> ::= <!-- The size of the OfficeArtBStoreContainerFileBlock
        array in bytes. -->

    <OfficeArtBStoreContainerFileBlock> ::= <OfficeArtFBSE> | <OfficeArtBlip>

    <OfficeArtBlip> ::= <OfficeArtBlipEMF> | <OfficeArtBlipWMF> |
        <OfficeArtBlipPICT> | <OfficeArtBlipJPEG> | <OfficeArtBlipPNG> |
        <OfficeArtBlipDIB> | <OfficeArtBlipTIFF>

    <OfficeArtBlipEMF> ::= <OfficeArtRecordHeader3> <rgbUid1> <rgbUid2>
        <OfficeArtMetafileHeader> <BlipFileData>

    <OfficeArtRecordHeader3> ::= <recVer3> <recInstance3> <recType3>
        <recLen3>

    <recVer3> ::= 0x0

    <recInstance3> ::= <!-- A value of 0x3D4 specifies one UID. A value of
        0x3D5 specifies two UIDs. -->

    <recType3> ::= 0xF01A

    <recLen3> ::= <!-- If recInstance3 is 0x3D4, then this value is equal to
        the size of BLIPFileData plus 50 bytes. If recInstance3 is 0x3D5,
        then this value is equal to the size of BLIPFileData plus 66 bytes
        00. -->

    <rgbUid1> ::= <!-- An MD4 digest that specifies the unique identifier of
```

```
        the uncompressed BLIPFileData. -->

    <rgbUid2> ::= <!-- An MD4 digest that specifies the unique identifier of
        the uncompressed BLIPFileData. If this value exists, then rgbUid1
        must be ignored. -->

    <OfficeArtMetafileHeader> ::= <cbSize> <rcBounds> <ptSize> <cbSave>
        <compression filter>

    <BLIPFileData> ::= <!-- Variable-length field that specifies BLIP (EMF,
        WMF) data. -->

    <OfficeArtBlipWMF> ::= <OfficeArtRecordHeader4> <rgbUid1> <rgbUid2>
        <OfficeArtMetafileHeader> <BlipFileData>

    <OfficeArtRecordHeader4> ::= <recVer4> <recInstance4> <recType4>
        <recLen4>

    <recVer4> ::= 0x0

    <recInstance4> ::= <!-- A value of 0x216 specifies one UID. A value of
        0x217 specifies two UIDs. -->

    <recType4> ::= 0xF01B

    <recLen4> ::= <!-- If recInstance4 is 0x216, then this value is equal to
        the size of BLIPFileData plus 50 bytes. If recInstance4 is 0x217,
        then this value is equal to the size of BLIPFileData plus 66 bytes
        00. -->
```

PRODUCT: WORD

GRAMMAR:

```
    <WordDocument Stream> ::= <OfficeArtContent>

    <OfficeArtContent> ::= <OfficeArtDggContainer> <OfficeArtWordDrawing>+
```

PRODUCT: EXCEL

GRAMMAR:

```
    <WORKSHEET> ::= <BOF> <WORKSHEETCONTENT>

    <WORKSHEETCONTENT> ::= [<Uncalced>] <Index> <GLOBALS> <PAGESETUP>
        [<HeaderFooter>] [<BACKGROUND>] *<BIGNAME> [<PROTECTION>] <COLUMNS>
        [<SCENARIOS>] <SORTANDFILTER> <Dimensions> [<CELLTABLE>] <OBJECTS>
        *<HFPicture> *<Note> *<PIVOTVIEW> [<DCON>] 1*<WINDOW> *<CUSTOMVIEW>
        *2<SORT> [<DxGCol>] *<MergeCells> [<LRng>] *<QUERYTABLE>
        [<PHONETICINFO>] <CONDFMTS> *<HLINK> [<DVAL>] [<CodeName>]
        *<WebPub> *<CellWatch> [<SheetExt>] *<FEAT> *<FEAT11> *<RECORD12>
        <EOF>

    <CHARTSHEET> ::= <BOF> <CHARTSHEETCONTENT>

    <CHARTSHEETCONTENT> ::= [<WriteProtect>] [<SheetExt>] [<WebPub>]
        *<HFPicture> <PAGESETUP> <PrintSize> [<HeaderFooter>]
        [<BACKGROUND>] *<Fbi> *<Fbi2> [<ClrtClient>] [<PROTECTION>]
        [<Palette>] [<SXViewLink>] [<PivotChartBits>] [<SBaseRef>]
```

```
        [<MsoDrawingGroup>] <OBJECTS> <Units> <CHARTFOMATS> <SERIESDATA>
        *<WINDOW> *<CUSTOMVIEW> [<CodeName>] [<CRTMLFRT>] <EOF>

<DIALOGSHEET> ::= <BOF> <DIALOGSHEETCONTENT>

<DIALOGSHEETCONTENT> ::= [<Uncalced>] <Index> <GLOBALS> <PAGESETUP>
        [<HeaderFooter>] *<BIGNAME> [<DIALOGPROTECTION>] <DefColWidth>
        <Dimensions> <OBJECTS> *<HFPicture> *<Note> 1*<DIALOGWINDOW>
        *<DIALOGCUSTOMVIEW> [<CodeName>] [<SheetExt>] *<RECORD12> <EOF>

<WORKBOOK> ::= <BOF> <WORKBOOKCONTENT>

<WORKBOOKCONTENT> ::= [<WriteProtect>] [<FilePass>] [<Template>]
        <INTERFACE> <WriteAccess> [<FileSharing>] <CodePage> *2047Lel DSF
        [Excel9File] RRTabId [ObProj] [ObNoMacros] [CodeName] [FNGROUPS]
        *Lbl [OleObjectSize] PROTECTION 1*Window1 Backup HideObj Date1904
        CalcPrecision RefreshAll BookBool FORMATTING
        *(PIVOTCACHEDEFINITION) [DOCROUTE] *UserBView UsesELFs
        1*BUNDLESHEET METADATA [MTRSettings] [ForceFullCalculation] Country
        *SUPBOOK *LBL *RTD [RecalcId] *HFPicture *MSODRAWINGGROUP
        [SHAREDSTRINGS] ExtSST *WebPub [WOpt] [CrErr] [BookExt] *FeatHdr
        *DConn [THEME] [CompressPictures] [Compat12] [GUIDTypeLib] EOF

<WORKBOOKCONTENT> ::= [<WriteProtect>] [<FilePass>] [<Template>]
        <INTERFACE> <WriteAccess> [<FileSharing>] <CodePage> *2047<Lel>
        <DSF> [<Excel9File>] <RRTabId> [<ObProj>] [<ObNoMacros>]
        [<CodeName>] [<FNGROUPS>] *<Lbl> [<OleObjectSize>] [<PROTECTION>]
        1*<Window1> <Backup> <HideObj> <DateObj> <Date1904> <CalcPrecision>
        <RefreshAll> <BookBool> <FORMATTING> *<PIVOTCACHEDEFINITION>
        [<DOCROUTE>] *<UserBView> <UsesELFs> 1*<BUNDLESHEET> <METADATA>
        [<MTRSettings>] [<ForceFullCalculation>] <Country> *<SUPBOOK>
        *<LBL> *<RTD> [<RecalcId>] *<HFPicture> *<MSODRAWINGGROUP>
        [<SHAREDSTRINGS>] <ExtSST> *<WebPub> [<WOpt>] [<CrErr>] [<BookExt>]
        *<FeatHdr> *<DConn> [<THEME>] [<CompressPictures>] [<Compat12>]
        [<GUIDTypeLib>] <EOF>

<MACROSHEET> ::= <BOF> <MACROSHEETCONTENT>

<MACROSHEETCONTENT> ::= [<Uncalced>] <Index> [<Intl>] <GLOBALS>
        <PAGESETUP> [<HeaderFooter>] [<BACKGROUND>] *<BIGNAME>
        [<PROTECTION>] <COLUMNS> <MACROSORTANDFILTER> <Dimensions>
        [<CELLTABLE>] <OBJECTS> *<HFPicture> *<Note> [<DCON>] 1*<WINDOW>
        *<CUSTOMVIEW> *2<SORT> [<DxGCol>] [<PHONETICINFO>] [<CodeName>]
        *<CellWatch> [<SheetExt>] *<FEAT> *<RECORD12> <EOF>

<HFPicture> ::= <FrtHeader> <unused> <reserved> [<OfficeArtDggContainer>
        | <OfficeArtDgContainer> ]

<MSODRAWINGGROUP> ::= <MsoDrawingGroup> *<Container>

<MsoDrawingGroup> ::= <OfficeArtDggContainer>
```

PRODUCT: POWERPOINT

GRAMMAR:

```
<DocumentContainer> ::= <RecordHeader> <DocumentAtom>
        <ExObjListContainer> <DocumentTextInfoContainer>
```

```
                <SoundCollectionContainer> <DrawingGroupContainer>
                <MasterListWithTextContainer> <DocInfoListContainer>
                <SlideHeadersFootersContainer> <NotesHeadersFootersContainer>
                <SlideListWithTextContainer> <NotesListWithTextContainer>
                <SlideShowDocInfoAtom> <NamedShowsContainer> <SummaryContainer>
                <DocRoutingSlipAtom> <PrintOptionsAtom>
                <RoundTripCustomTableStyles12Atom> <EndDocumentAtom>
                <rtCustomTableStylesAtom2>

        <DrawingGroupContainernp> ::= <RecordHeader> <OfficeArtDggContainer>
```

OFFICE 2003.4.19: END

OFFICE 2003.4.20: Database Connections and Queries

PRODUCT: WORD

GRAMMAR:

```
        <FibRgFcLcb97> ::= <Octect>[348] <fcPms> <lcbPms> <Octect>[384]

        <fcPms> ::= <!-- An unsigned integer that specifies an offset in the
             Table Stream. A Pms, which contains the current state of a print
             merge operation, begins at this offset. If lcbPms is zero, then
             fcPms is undefined and MUST be ignored. -->

        <lcbPms> ::= <!-- An unsigned integer which specifies the size, in bytes,
             of the Pms at offset fcPms. -->

        <FibRgFcLcb2002> ::= <Octect>[1004] <fcPmsNew> <lcbPmsNew> <fcODSO>
             <lcbODSO> <Octect>[64]

        <fcPmsNew> ::= <!-- An unsigned integer that specifies an offset in the
             Table Stream. A new Pms, which contains the current state of a
             print merge operation, begins at this offset. If lcbPmsNew is zero,
             then fcPmsNew is undefined and MUST be ignored. -->

        <lcbPmsNew> ::= <!-- An unsigned integer which specifies the size, in
             bytes, of the Pms at offset fcPmsNew. -->

        <fcODSO> ::=  <!-- An unsigned integer that specifies an offset in the
             Table Stream. Office Data Source Object (ODSO) data used to perform
             mail merge begins at this offset. The data is stored in an array of
             ODSOPropertyBase items. The ODSOPropertyBase items are of variable
             size and are stored contiguously. The complete set of properties
             contained in the array is determined by reading each
             ODSOPropertyBase, until a total of lcbODSO bytes have been read. If
             lcbODSO is zero, then fcODSO is undefined and MUST be ignored. -->

        <lcbODSO> ::= <!-- An unsigned integer that specifies the size, in bytes,
             of the Office Data Source data at offset fcODSO in the Table
```

```
        Stream. -->

    <Pms> ::= <wpms> <ipmfMF> <ipmfFetch> <iRecCur> <rgpmfs> <rfs>
        <cblszSqlStr> <lxszSqlStr> <sttbfRfs> <wpmsdt>

    <wpms> ::= <!-- The mail merge state as a Wpms. See [MS-DOC] 2.9.344 -->

    <ipmfMF> ::= <!-- An unsigned integer that specifies the index in the
        array rgpmfs and MUST be 0 or 1. This is for the mail merge header
        field source, from where the mail merge column names are obtained.
        -->

    <ipmfFetch> ::= <!-- An unsigned integer that specifies the index in the
        array rgpmfs and MUST be 0 or 1. This is for the mail merge data
        fetch source, from where the mail merge data are obtained. -->

    <iRecCur> ::= <!-- An unsigned integer that specifies the index of the
        current mail merge record. iRecCur MUST be either between 0 and
        0xFFFFFFF0 as the record index, or 0xFFFFFFFF as a nil value. -->

    <rgpmfs> ::= <Pmfs> <Pmfs>

    <Pmfs> ::= <!-- A Pmfs structure specifies the mail merge data source
        connection properties. See Pmfs in [MS-DOC] 2.9.201. -->

    <rfs> ::= <!-- The mail merge record filtering information. See Rfs in
        [MS-DOC] 2.9.224. -->

    <cblszSqlStr> ::= <!-- An unsigned integer that specifies the length, in
        number bytes, of the string lxszSqlStr. Because lxszSqlStr is in
        Unicode, cblszSqlStr MUST be an even number. If cblszSqlStr is
        zero, lxszSqlStr does not exist, otherwise cblszSqlStr MUST be
        larger than 2 but not larger than 512. -->

    <lxszSqlStr> ::= <!-- The null-terminated Unicode SQL Query string. For
        example, “SELECT * FROM [myTable] WHERE …”, where myTable is the
        table name in the database connected. lxszSqlStr is not present if
        cblxszSqlStr is zero. -->

    <sttbfRfs> ::= <!-- The string table, STTB, that contains the string for
        mail merge connection and record filtering. See SttbfRfs in [MS-
        DOC] 2.9.286. Pms.sttbfRfs does not exist if Pms.rfs.hsttbRfs is
        zero. See Rfs in [MS-DOC] 2.9.224. -->

    <wpmsdt> ::= <!-- The mail merge document type. See Wpmsdt in [MS-DOC]
        2.9.345. -->
```

PRODUCT: EXCEL

GRAMMAR:

```
    <WORKBOOKCONTENT> ::= [<WriteProtect>] [<FilePass>] [<Template>]
        <INTERFACE> <WriteAccess> [<FileSharing>] <CodePage> *2047<Lel>
        <DSF> [<Excel9File>] <RRTabId> [<ObProj>] [<ObNoMacros>]
        [<CodeName>] [<FNGROUPS>] *<Lbl> [<OleObjectSize>] [<PROTECTION>]
        1*<Window1> <Backup> <HideObj> <DateObj> <Date1904> <CalcPrecision>
        <RefreshAll> <BookBool> <FORMATTING> *<PIVOTCACHEDEFINITION>
        [<DOCROUTE>] *<UserBView> <UsesELFs> 1*<BUNDLESHEET> <METADATA>
```

```
        [<MTRSettings>] [<ForceFullCalculation>] <Country> *<SUPBOOK>
        *<LBL> *<RTD> [<RecalcId>] *<HFPicture> *<MSODRAWINGGROUP>
        [<SHAREDSTRINGS>] <ExtSST> *<WebPub> [<WOpt>] [<CrErr>] [<BookExt>]
        *<FeatHdr> *<DConn> [<THEME>] [<CompressPictures>] [<Compat12>]
        [<GUIDTypeLib>] <EOF>

<PIVOTCACHEDEFINITION> ::= <SXStreamID> <SXVS> [<SXSRC>] [<SXADDLCACHE>]

<SXStreamID> ::= <!-- An unsigned two-byte integer that specifies a
     stream in the PivotCache storage. -->

<SXVS> ::= <!-- A value of 0x002 represents an External connection and
     that the value of SXSRC will be a DbQuery. -->

<SXSRC> ::= <DREF> | <SXTBL> | <DBQUERY>

<DBQUERY> ::= <DbOrParamQry> [1*<SXString> [<DbOrParamQry> *(<SXString>
     <DbOrParamQry>)]] *<SXString>

<DbOrParamQry> ::= <DbQuery> | <ParamQry> <!-- This record specifies a
     DbQuery or ParamQry record depending on the record that precedes
     this record. -->

<DbQuery> ::= <!-- This record specifies information about an external
     connection. This record is followed by SXString and ParamQry
     records that specify the strings and parameters. See [MS-XLS].pdf
     Section 2.4.80 for a complete specification. -->

<ParamQry> ::= <!-- This record specifies the parameters for a
     parameterized query. See [MS-XLS].pdf Section 2.4.190 for a
     complete specification. -->

<SXString> ::= <!-- This record specifies a segment of a string that
     contains information about a PivotCache or an external connection.
     When preceeded by the DbQuery, it can contain an SQL Query string.
     See [MS-XLS].pdf Section 2.4.304 for a complete specification. -->

<SXADDLCACHE> ::= <SXAddl_SXCCache_SXDId> <SXAddl_SXCCache_SXDVer10Info>
     [<SXAddl_SXCCache_SXDVerSXMacro>] [<SXADDLCACHE12>]
     [<SXADDLDBQUERY>] *<UNKNOWNFRT> <SXAddl_SXCCache_SXDEnd>

<SXADDLDBQUERY> ::= [<SXAddl_SXCQuery_SXDMLSource>
     *<Continue_SxaddlSxString>] [<SXAddl_SXCQuery_SXDSrcDataFile>
     *<Continue_SxaddlSxString>] [<SXAddl_SXCQuery_SXDSrcConnFile>
     *<Continue_SxaddlSxString>] [<SXAddl_SXCQuery_SXDSrcReconnCond>]
     <SXAddl_SXCQuery_SXDEnd> <!-- This record specifies additional
     connection information for a PivotTable view, PivotCache, or query
     table. -->

<DConn> ::= <!-- This record specifies information for a single data
     connection. This record will contain paths to database files or
     servers as well as username and password data. It can also contain
     a database SQL commands. See [MS-XLS].pdf Section 2.4.84 for a
     complete specification. -->

<WORKSHEETCONENT> ::= [<Uncalced>] <Index> <GLOBALS> <PAGESETUP>
     [<HeaderFooter>] [<BACKGROUND>] *<BIGNAME> [<PROTECTION>] <COLUMNS>
```

```
            [<SCENARIOS>] <SORTANDFILTER> <Dimensions> [<CELLTABLE>] <OBJECTS>
            *<HFPicture> *<Note> *<PIVOTVIEW> [<DCON>] 1*<WINDOW> *<CUSTOMVIEW>
            *2<SORT> [<DxGCol>] *<MergeCells> [<LRng>] *<QUERYTABLE>
            [<PHONETICINFO>] <CONDFMTS> *<HLINK> [<DVAL>] [<CodeName>]
            *<WebPub> *<CellWatch> [<SheetExt>] *<FEAT> *<FEAT11> *<RECORD12>
            <EOF>

      <QUERYTABLE> ::= <Qsi> <DBQUERY> <QsiSXTag> <DBQUERYEXT> [<SXADDLQSI>]
            [<QSIR>] [<SORTDATA12>]

      <Qsi> ::= <!-- This record specifies properties for a query table. See
            [MS-XLS].pdf Section 2.4.208 for a complete specification. -->

      <QsiSXTag> ::= <!-- This record specifies the name and refresh
            information for a query table or a PivotTable view. See [MS-
            XLS].pdf Section 2.4.211 for a complete specification. -->

      <SXADDLQSI> ::= <SXAddl_SXCQsi_SXDId> <SXADDLDBQUERY> *<UNKNOWNFRT>
            <SXAddl_SXCQsi_SXDEnd>

      <DBQUERYEXT> ::= <DBQueryExt> [<ExtString>] *4[<OleDbConn> *<ExtString>]
            [<TxtQry> *<ExtString>]

      <DBQueryExt> ::= <!-- This record specifies information about an external
            connection. See [MS-XLS].pdf Section 2.4.81 for a complete
            specification. -->

      <ExtString> ::= <!-- This record specifies the connection string for a
            query that retrieves external data. See [MS-XLS].pdf Section
            2.4.108 for a complete specification. -->

      <OleDbConn ::= <!--  This record specifies the connection information for
            an OLE DB connection string, and specifies the beginning of a
            collection of ExtString records that specifies the connection
            string for a query that retrieves external data. See [MS-XLS].pdf
            Section 2.4.186 for a complete specification. -->

      <TxtQry> ::= <!-- This record specifies connection information for an
            external text query. See [MS-XLS].pdf Section 2.4.330 for a
            complete specification. -->
```

OFFICE 2003.4.20: END

OFFICE 2003.4.21: Tracked Changes (N/A)

OFFICE 2003.4.21: END

OFFICE 2003.4.22: Versions (N/A)

OFFICE 2003.4.22: END

OFFICE 2003.4.23: FastSave Data (N/A)

OFFICE 2003.4.23: END

OFFICE 2003.4.24: Document Protections

PRODUCT: WORD

GRAMMAR

```
<FibBase> ::= <wIdent> <nFib> <unused> <lid> <pnNext> <fDot> <fGlsy>
     <fComplex> <fHasPic> <cQuickSaves> <fEncrypted> <fWhichTblStm>
     <fReadOnlyRecommended> <fWriteReservation> <fExtChar>
     <fLoadOverride> <fFarEast> <fObfuscated> <nFibBack> <IKey>
     <EncryptionHeader> <envr> <fMac> <fEmptySpecial>
     <fLoadOverridePage> <reserved1> <reserved2> <fSpare> <reserved3>
     <reserved4> <reserved5> <reserved6>
```

Write Reservation Password:

```
     <SttbfAssoc> ::= <fExtend> <cData> <dbExtra> *( <cchData> <Data> )

     <fExtend> ::= 0xFFFF

     <cData> ::= 0x0012

     <cbExtra> ::= 0

<cchData> ::= 0x11 <!-- Index signifying write-reservation password for
     document. -->

     <Data> ::= <!-- 15 characters or less password -->
```

Encryption:

```
<RC4 CryptoAPI Encrypted Summary Stream> ::=
     <StreamDescriptorArrayOffset> <StreamDescriptorArraySize>
     <EncryptedStreamData> <EncryptedStreamDescriptorArray>

<EncryptedStreamDescriptorArray> ::= *<EncryptedStreamDescriptor>

<EncryptedStreamDescriptor> ::= <StreamOffset> <StreamSize> <Block>
     <NameSize> <fStream> <Reserved1> <Unused> <Reserved2> <StreamName>

<EncryptionHeader> ::= <Flags> <SizeExtra> <AlgID> <AlgIDHash> <KeySize>
     <ProviderType> <Reserved1> <Reserved2> <CSPName>

<Flags> ::= <Reserved1> <Reserved2> <fCryptoAPI> <fDocProps> <fExternal>
     <fAES> <Unused>
```

See [MS-OFFCRYPTO].pdf, Section 2.3, Encryption.

Digital Signatures:

```
<_signatures> ::= <CryptoAPI Digital Signature Structure>

<CryptoAPI Digital Signature Structure> ::= <CertificateSize>
     <ItermediateCertificatesStore> <CertificateInfoArray> <EndMarker>

<CertificateInfoArray> ::= *<CryptoAPI Digital Signature CertificateInfo
     Structure>

<CryptoAPI Digital Signature CertificateInfo Structure> ::=
     <CertificateInfoSize> <SignerLength> <IssuerLength> <ExpireTime>
     <SignTime> <AlgIDHash> <SignatureSize> <EncodedCertificateSize>
     <Version> <SerialNumberSize> <IssuerBlobSize> <Reserved>
     <SignerName> <IssuerName> <Signature> <EncodedCertificate>
     <SerialNumber> <IssuerBlob>

<ExpireTime> ::= <HighDateTime> <LowDateTime>
```

See [MS-OFFCRYPTO].pdf, Section 2.5.1,CryptoAPI Digital Signature Structures and Streams.

```
<_xmlsignatures> ::= *<XMLDSig>

<XMLDSig> ::= <!-- See [MS-OFFCRYPTO].pdf, Section 2.5.2, Xmldsig Digital
     Signature Elements for more details. -->

[MS-OFFCRYPTO].pdf, Section 2.5.3, _xmlsignatures Storage
```

PRODUCT: EXCEL

GRAMMAR

Write Reservation Password:

```
<DIALOGSHEETCONTENT> ::= [<Uncalced>] <Index> <GLOBALS> <PAGESETUP>
     [<HeaderFooter>] *<BIGNAME> [<DIALOGPROTECTION>] <DefColWidth>
     <Dimensions> <OBJECTS> *<HFPicture> *<Note> 1*<DIALOGWINDOW>
     *<DIALOGCUSTOMVIEW> [<CodeName>] [<SheetExt>] *<RECORD12> <EOF>


     <DIALOGPROTECTION> ::= <Protect> <Password>

<CHARTSHEET> ::= <BOF> <CHARTSHEETCONTENT>

<CHARTSHEETCONTENT> ::= [<WriteProtect>] [<SheetExt>] [<WebPub>]
     *<HFPicture> <PAGESETUP> <PrintSize> [<HeaderFooter>]
     [<BACKGROUND>] *<Fbi> *<Fbi2> [<ClrtClient>] [<PROTECTION>]
     [<Palette>] [<SXViewLink>] [<PivotChartBits>] [<SBaseRef>]
     [<MsoDrawingGroup>] <OBJECTS> <Units> <CHARTFOMATS> <SERIESDATA>
     *<WINDOW> *<CUSTOMVIEW> [<CodeName>] [<CRTMLFRT>] <EOF>

<WORKBOOK> ::= <BOF> <WORKBOOKCONTENT>

<WORKBOOKCONTENT> ::= [<WriteProtect>] [<FilePass>] [<Template>]
     <INTERFACE> <WriteAccess> [<FileSharing>] <CodePage> *2047<Lel>
     <DSF> [<Excel9File>] <RRTabId> [<ObProj>] [<ObNoMacros>]
     [<CodeName>] [<FNGROUPS>] *<Lbl> [<OleObjectSize>] [<PROTECTION>]
     1*<Window1> <Backup> <HideObj> <DateObj> <Date1904> <CalcPrecision>
     <RefreshAll> <BookBool> <FORMATTING> *<PIVOTCACHEDEFINITION>
     [<DOCROUTE>] *<UserBView> <UsesELFs> 1*<BUNDLESHEET> <METADATA>
```

```
      [<MTRSettings>] [<ForceFullCalculation>] <Country> *<SUPBOOK>
      *<LBL> *<RTD> [<RecalcId>] *<HFPicture> *<MSODRAWINGGROUP>
      [<SHAREDSTRINGS>] <ExtSST> *<WebPub> [<WOpt>] [<CrErr>] [<BookExt>]
      *<FeatHdr> *<DConn> [<THEME>] [<CompressPictures>] [<Compat12>]
      [<GUIDTypeLib>] <EOF>

<MACROSHEET> ::= <BOF> <MACROSHEETCONTENT>

<MACROSHEETCONTENT> ::= [<Uncalced>] <Index> [<Intl>] <GLOBALS>
     <PAGESETUP> [<HeaderFooter>] [<BACKGROUND>] *<BIGNAME>
     [<PROTECTION>] <COLUMNS> <MACROSORTANDFILTER> <Dimensions>
     [<CELLTABLE>] <OBJECTS> *<HFPicture> *<Note> [<DCON>] 1*<WINDOW>
     *<CUSTOMVIEW> *2<SORT> [<DxGCol>] [<PHONETICINFO>] [<CodeName>]
     *<CellWatch> [<SheetExt>] *<FEAT> *<RECORD12> <EOF>

<WORKSHEET> ::= <BOF> <WORKSHEETCONTENT>

<WORKSHEETCONENT> ::= [<Uncalced>] <Index> <GLOBALS> <PAGESETUP>
     [<HeaderFooter>] [<BACKGROUND>] *<BIGNAME> [<PROTECTION>] <COLUMNS>
     [<SCENARIOS>] <SORTANDFILTER> <Dimensions> [<CELLTABLE>] <OBJECTS>
     *<HFPicture> *<Note> *<PIVOTVIEW> [<DCON>] 1*<WINDOW> *<CUSTOMVIEW>
     *2<SORT> [<DxGCol>] *<MergeCells> [<LRng>] *<QUERYTABLE>
     [<PHONETICINFO>] <CONDFMTS> *<HLINK> [<DVAL>] [<CodeName>]
     *<WebPub> *<CellWatch> [<SheetExt>] *<FEAT> *<FEAT11> *<RECORD12>
     <EOF>

<PROTECTION> ::= [<Protect>] [<ScenarioProtect>] [<ObjProtect>]
     [<Password>]

<WinProtect> ::= 0x0000 | 0x0001 <!-- A Boolean that specifies whether
     the workbook windows can be resized or moved and whether the window
     state can be changed. See [MS-XLS].pdf, Section 2.4.347 for
     details.-->

<Protect> ::= 0x0000 | 0x0001 <!-- A Boolean that specifies whether the
     sheet or workbook is protected. See [MS-XLS].pdf, Section 2.4.207
     for details. -->

<Password> ::= <!-- An unsigned integer that specifies the password
     verifier<95>. See Password Verifier Algorithm for more information.
     If the password is for a sheet, MUST NOT equal 0x0000. If wPassword
     is 0x0000 it means the workbook has no password. See [MS-XLS].pdf,
     Section 2.4.191 for details. -->

<Prot4Rev> ::= 0x0000 | 0x0001 <!-- A Boolean that specifies whether
     removal of the the shared workbook's revision logs is disallowed.
     See [MS-XLS].pdf, Section 2.4.205 for details. -->

<Prot4RevPass> ::= <!-- An unsigned integer that specifies the password
     verifier that is required to change the value of the Prot4Rev
     record that immediately precedes this record. See [MS-XLS].pdf,
     Section 2.4.206 for details. -->

<ScenarioProtect> ::= 0x0000 | 0x0001 <!-- A Boolean that specifies
     whether the scenarios in the sheet are protected. See [MS-XLS].pdf,
     Section 2.4.245 for details. -->
```

```
<ObjProtect> ::= 0x0001 <!-- A Boolean that specifies that the objects
     are protected. MUST be 0x0001. See [MS-XLS].pdf, Section 2.4.183
     for details. -->

Encryption:

<FilePass> ::= <wEncryptionType> <encryptionInfo>

<wEncryptionType> ::= 0x0000 | 0x0001 <!-- A Boolean that specifies the
     encryption type. See [MS-XLS].pdf, Section 2.4.117 for details.  --
     >

<encryptionInfo> ::= 0x0001 | 0x0002 | 0x0003 <!-- A Boolean that
     specifies the encryption type. See [MS-XLS].pdf, Section 2.4.117
     for details.  -->

<RC4 CryptoAPI Encrypted Summary Stream> ::=
     <StreamDescriptorArrayOffset> <StreamDescriptorArraySize>
     <EncryptedStreamData> <EncryptedStreamDescriptorArray>

<EncryptedStreamDescriptorArray> ::= *<EncryptedStreamDescriptor>

<EncryptedStreamDescriptor> ::= <StreamOffset> <StreamSize> <Block>
     <NameSize> <fStream> <Reserved1> <Unused> <Reserved2> <StreamName>

<EncryptionHeader> ::= <Flags> <SizeExtra> <AlgID> <AlgIDHash> <KeySize>
     <ProviderType> <Reserved1> <Reserved2> <CSPName>

<Flags> ::= <Reserved1> <Reserved2> <fCryptoAPI> <fDocProps> <fExternal>
     <fAES> <Unused>
```

See [MS-OFFCRYPTO].pdf, Section 2.3.2, Encryption.

Digital Signatures:

```
<_signatures> ::= <CryptoAPI Digital Signature Structure>

<CryptoAPI Digital Signature Structure> ::= <CertificateSize>
     <ItermediateCertificatesStore> <CertificateInfoArray> <EndMarker>

<CertificateInfoArray> ::= *<CryptoAPI Digital Signature CertificateInfo
     Structure>

<CryptoAPI Digital Signature CertificateInfo Structure> ::=
     <CertificateInfoSize> <SignerLength> <IssuerLength> <ExpireTime>
     <SignTime> <AlgIDHash> <SignatureSize> <EncodedCertificateSize>
     <Version> <SerialNumberSize> <IssuerBlobSize> <Reserved>
     <SignerName> <IssuerName> <Signature> <EncodedCertificate>
     <SerialNumber> <IssuerBlob>

<ExpireTime> ::= <HighDateTime> <LowDateTime>
```

See [MS-OFFCRYPTO].pdf, Section 2.5.1, CryptoAPI Digital Signature Structures and Streams.

```
<_xmlsignatures> ::= *<XMLDSig>

<XMLDSig> ::= <!-- See [MS-OFFCRYPTO].pdf, Section 2.5.2, Xmldsig Digital
```

```
        Signature Elements for more details. -->
```

See [MS-OFFCRYPTO].pdf, Section 2.5.3, _xmlsignatures Storage.

PRODUCT: POWERPOINT

GRAMMAR

Write Reservation Password:

```
<PowerPointDocument Stream> ::= <DocumentContainer> |
     <MasterOrSlideContainer> | <HandoutContainer> | <SlideContainer> |
     <NotesContainer> | <ExOleObjStg> | <ExControlStg> | <VbaProjectStg>
     | <PersistDirectoryAtom> | <UserEditAtom>

<DocumentContainer> ::= <rh> <documentAtom> <exObjList>
     <documentTextInfo> <soundCollection> <drawingGroup> <masterList>
     <docInfoList> <slideHF> <notesHF> <slideList>
     <slideShowDocInfoAtom> <namedShows> <summary> <docRoutingSlipAtom>
     <printOptionsAtom> <rtCustomTableStylesAtom1>
     <rtCustomTableStylesAtom2>

<docInfoList> ::= <DocInfoListContainer>

<DocInfoListContainer> ::= <rh> <DocInfoListSubContainerOrAtom>

<DocInfoListSubContainerOrAtom> ::= <DocProgTagsContainer> |
     <NormalViewSetInfoContainer> | <NotesTextViewInfoContainer> |
     <OutlineViewInfoContainer> | <SlideViewInfoInstance> |
     <SorterViewInfoContainer> | <VBAInfoContainer>

<DocProgTagsContainer> ::= <rh> <rgChildRec>

<rgChildRec> ::= *<DocProgTagsSubContainerOrAtom>

<DocProgTagsSubContainerOrAtom> ::= <ProgStringTagContainer> |
     <DocProgBinaryTagContainer>

<DocProgBinaryTagContainer> ::= <rh> <DocProgBinaryTagSubContainerOrAtom>

<DocProgBinaryTagSubContainerOrAtom> ::= <PP9DocBinaryTagExtension> |
     <PP10DocBinaryTagExtension> | <PP11DocBinaryTagExtension> |
     <PP12DocBinaryTagExtension> | <UnknownBinaryTag>

<PP10DocBinaryTagExtension> ::= <rh> <tagName> <rhData>
     <fontCollectionContainer> <rgTextMasterStyle10> <textDefaultsAtom>
     <gridSpacingAtom> <rgCommentIndex10> <fontEmbedFlagsAtom>
     <copyrightAtom> <keywordsAtom> <filterPrivacyFlagsAtom>
     <outlineTextPropsContainer> <docToolbarStatesAtom>
     <slideListTableContainer> <rgDiffTree10Container>
     <modifyPasswordAtom> <photoAlbumInfoAtom>

     <ModifyPasswordAtom> ::= <rh> <modifyPassword>

<modifyPassword> ::= <!-- A PrintableUnicodeString that specifies a
     password used to modify the document. See [MS-PPT].pdf, Section
     2.4.7. -->
```

Encryption:

```
<EncryptedSummary> ::= <StreamDescriptorArrayOffset>
     <StreamDescriptorArraySize> <EncryptedStreamData>
     <EncryptedStreamDescriptorCount> <EncryptedStreamDescriptorArray>

<EncryptedStreamDescriptorArray> ::= *<EncryptedStreamDescriptor>

<EncryptedStreamDescriptor> ::= <StreamOffset> <StreamSize> <Block>
     <NameSize> <fStream> <Reserved1> <Unused> <Reserved2> <StreamName>

---

<CryptSession10Container> ::= <rh> <data>

<rh> ::= 0xF 0x000 <RT_CryptSession10Container>

<data> ::= <EncryptionHeader>

<EncryptionHeader> ::= <EncryptionVersionInfo> <EncryptionHeader.Flags>
     <EncryptionHeaderSize> <EncryptionHeader2> <EncryptionVerifier>

<EncryptionHeader2> ::= <Flags> <SizeExtra> <AlgID> <AlgIDHash> <KeySize>
     <ProviderType> <Reserved1> <Reserved2> <CSPName>

<Flags> ::= <Reserved1> <Reserved2> <fCryptoAPI> <fDocProps> <fExternal>
     <fAES> <Unused>

<EncryptionVerifier> ::= <SaltSize> <Salt> <EncryptedVerifier>
     <VerifierHashSize> <EncryptedVerifierHash>

---

<Current User Stream> ::= <CurrentUserAtom>

<CurrentUserAtom> ::= <rh> <size> <headerToken> <offsetToCurrentEdit>
     <lenUserName> <docFileVersion> <majorVersion> <minorVersion>
     <unused> <ansiUserName> <relVersion> <unicodeUserName>

<headerToken> ::= 0xE391C05F | 0xF3D1C4DF <!-- An unsigned integer that
     specifies a token used to identify whether the file is encrypted.
     0xE391C05F means the file should not be encrypted. 0xF3D1C4DF means
     the file MUST be an ecrypted document. -->

<PowerPointDocument Stream> ::= <DocumentContainer> |
     <MasterOrSlideContainer> | <HandoutContainer> | <SlideContainer> |
     <NotesContainer> | <ExOleObjStg> | <ExControlStg> | <VbaProjectStg>
     | <PersistDirectoryAtom> | <UserEditAtom>

<UserEditAtom> ::= <rh> <lastSlideIdRef> <version> <minorVersion>
     <majorVersion> <offsetLastEdit> <offsetPersistDirectory>
     <docPersistIdRef> <persistIdSeed> <lastView> <unused>
     <encryptSessionPersistIdRef>

<encryptSessionPersistIdRef> ::= <!-- An optional PersistIdRef that
     specifies the value to look up in the persist object directory to
```

```
                find the offset of the CryptSession10Container record. It MAY<11>
                be omitted. It MUST exist if the document is an encrypted document.
                -->
```

Digital Signatures:

```
        <_signatures> ::= <CryptoAPI Digital Signature Structure>

        <CryptoAPI Digital Signature Structure> ::= <CertificateSize>
             <ItermediateCertificatesStore> <CertificateInfoArray> <EndMarker>

        <CertificateInfoArray> ::= *<CryptoAPI Digital Signature CertificateInfo
             Structure>

        <CryptoAPI Digital Signature CertificateInfo Structure> ::=
             <CertificateInfoSize> <SignerLength> <IssuerLength> <ExpireTime>
             <SignTime> <AlgIDHash> <SignatureSize> <EncodedCertificateSize>
             <Version> <SerialNumberSize> <IssuerBlobSize> <Reserved>
             <SignerName> <IssuerName> <Signature> <EncodedCertificate>
             <SerialNumber> <IssuerBlob>

        <ExpireTime> ::= <HighDateTime> <LowDateTime>

        [MS-OFFCRYPTO].pdf, Section 2.5.1,CryptoAPI Digital Signature Structures
             and Streams

        <_xmlsignatures> ::= *<XMLDSig>

        <XMLDSig> ::= <!-- See [MS-OFFCRYPTO].pdf, Section 2.5.2, Xmldsig Digital
             Signature Elements for more details. -->
```

See [MS-OFFCRYPTO].pdf, Section 2.5.3, _xmlsignatures Storage.

OFFICE 2003.4.24: END

OFFICE 2003.4.25: Color & Size Obfuscation (N/A)

OFFICE 2003.4.25: END