



***Inspection and Sanitization
Guidance
for Microsoft Office 2007 and Office Open XML
(OOXML)***

Version 1.0

18 June 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
06/18/2010	1.0	Initial Release
12/13/2017	1.0	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 2007* document provides guidance and specifications for developing file inspection and sanitization software for Microsoft®¹ (MS) Office 2007 files (i.e., word processing, presentation, and spreadsheet documents).

Client programs, such as MS Office Word® 2007, can store arbitrary data in a MS Office 2007 document, including video, sounds, and hidden text. This presents a challenge for automated file processing software. This document addresses this challenge by delineating the various constructs with the Office 2007 file formats. The document provides specific guidelines for designing, building, and testing Office 2007 file inspection and sanitization applications.

¹ Microsoft Office and Word are registered trademarks of Microsoft Corp.

TABLE OF CONTENTS

- 1. SCOPE..... 1-1**
 - 1.1 PURPOSE OF THIS DOCUMENT 1-1
 - 1.2 INTRODUCTION..... 1-1
 - 1.3 BACKGROUND..... 1-2
 - 1.4 DOCUMENT ORGANIZATION..... 1-4
 - 1.5 RECOMMENDATIONS 1-5
 - 1.5.1 Actions..... 1-5
 - 1.5.2 Action Options 1-6
 - 1.5.3 Naming Convention for Recommendations..... 1-7
 - 1.6 DTG 1-7
 - 1.7 DOCUMENT LIMITATIONS..... 1-8
 - 1.7.1 Constructs 1-8
 - 1.7.2 Covert Channel Analysis..... 1-10
 - 1.7.3 Character Encoding 1-10
- 2. CONSTRUCT AND TAXONOMY OVERVIEW 2-1**
 - 2.1 CONSTRUCTS 2-1
 - 2.2 TAXONOMY 2-1
- 3. OFFICE 2007 FILE STRUCTURE OVERVIEW 3-1**
 - 3.1 OFFICE 2007 OOXML..... 3-1
 - 3.2 EMBEDDED AND LINKED OBJECTS 3-3
 - 3.3 OFFICE 2007 ZIP FILE STRUCTURE AND FORMAT 3-6
 - 3.4 MACROS AND MALICIOUS CONTENT OVERVIEW..... 3-18
 - 3.4.1 Font Embedding in Office Documents..... 3-19
- 4. OFFICE 2007 CONSTRUCTS AND METADATA 4-1**
- 5. XML SCHEMA CONSTRUCTS 5-1**
- 6. ACRONYMS 6-1**
- 7. REFERENCED DOCUMENTS A-3**
- APPENDIX A: BNF GRAMMAR..... A-5**

LIST OF FIGURES

Figure 3-1 Word Format..... 3-2

Figure 3-2: Excel² Format..... 3-2

Figure 3-3: PowerPoint[®] Format..... 3-3

Figure 3-4: Embedded Content 3-5

Figure 3-5 Zip File Format..... 3-7

² Excel and PowerPoint are registered trademarks of Microsoft Corp.

LIST OF TABLES

Table 1-1. Document Organization..... 1-4

Table 1-2. Recommendation Actions 1-6

Table 1-3. Recommendation Action Options 1-7

Table 2-1 Document Taxonomy 2-2

Table 3-1: Office 2007 Extensions^[10] 3-19

Table 6-1 Acronyms..... 6-1

1. SCOPE

1.1 Purpose of this Document

This *Inspection and Sanitization Guidance (ISG) for Microsoft Office 2007* document outlines the findings of the Cross Domain Solutions (CDS) team's research into the potential areas of concern within the Office Open XML (OOXML) format that Microsoft® (MS) Corporation uses in their MS Office 2007 product suite (i.e., Excel, PowerPoint and Word).

The audience for this document includes system engineers, designers, software developers, and testers who work on file inspection and sanitization applications that process MS Office 2007 files.

1.2 Introduction

The frequent exchange of MS Office documents between federal government entities and business partners creates a significant potential risk for sensitive information release. In this context, sensitive information may fall into two general categories: unintentional release of data and intentional obfuscation of data to hide information in the file. This document focuses on the accidental release perspective, although the sanitization guidance outlined herein will also aid in identifying and removing intentionally hidden content.

Microsoft's introduction of the Office Open XML (OOXML) file format as the default Office 2007 storage mechanism creates an opportunity to develop tools to inspect and sanitize hidden data, metadata, and malicious code more effectively. Past development efforts sought to leverage open standards, such as eXtensible Markup Language (XML), for this purpose. However, these efforts required in-depth knowledge of the existing binary Object Linking and Embedding (OLE) Compound Document Format (CDF)-based (i.e., DocFile) that was difficult or impossible to obtain. In addition, converting files back to DocFile format, following sanitization, reintroduced the potential for hidden data or metadata. It also made it difficult to evaluate sanitization success. (Hereafter, this document refers to DocFile-based binaries as pre-OOXML binary formats.)

Creating the OOXML specification, and its underlying schemas, helps ensure an accurate representation of all available features in the XML files and the content (within version compatibility limitations). In addition to the native support in Office 2007 and higher versions, Office 2000, XP, and 2003 have the capability to read and write

OOXML with the Compatibility Pack installed. This plug-in is a free download from Microsoft's Web site. Network administrators also have the option of making OOXML the default *Save* format (for previous versions with the Compatibility Pack installed) in environments that encourage XML format use.

The format's openness offers increased opportunities for tool development. It also enables a more thorough evaluation of sanitization tools to determine if they effectively achieve the stated inspection and sanitization goals. This document identifies the minimum set of inspection and sanitization capabilities that a tool should perform to offer reasonable assurance and effective mitigation of hidden data, metadata, and malicious code risks to more safely transfer Office documents between parties.

OOXML's enhanced inspection and sanitization capabilities do not mean that one should exchange files in OOXML or any other compound document format by default.

1.3 Background

The need to parse MS Office pre-OOXML binary documents led to the development of many approaches for modifying and extracting data from these file types. The most notable are Office Automation tools, developed by Tonic Systems (now part of Google³), Apache⁴ Poor Obfuscated Implementation (POI), OpenOffice.org XML conversion and native bit parsing. This section briefly describes background information about historical difficulties with parsing Office files that are stored in the pre-OOXML formats.

Microsoft provides an Application Programming Interface (API) that enables developers to leverage development environments, such as Visual Studio, to control Office programmatically. A significant benefit of opening Office files in their native applications is the reduced need to know the internal file structure. The drawbacks of this approach include a lack of knowledge about potential remaining hidden or embedded data, performance constraints, Windows platform dependence, likelihood of program instability when opening large or malformed files, and functionality limitation based upon the API capabilities.

Developers have created a many applications to modify or convert pre-OOXML Office files. Tonic Systems produced one of the tools that are commonly used for applications, such as Google Docs. When Google acquired Tonic Systems in 2007, Google incorporated their Office capabilities for creating and converting docs, and supporting

³ Google is a registered trademark of Google, Inc.

⁴ Apache is a registered trademark of the Apache Software Foundation

collaboration.^[1] An example of another toolset is the Apache POI project, which the Apache Software Foundation runs. The POI suite of tools also supports creating and modifying pre-OOXML Office files (with OOXML format support currently in development).

OpenOffice.org enables converting pre-OOXML binary Office files to the Organization for the Advancement of Structured Information Standards (OASIS) Open Document Format (ODF), a zip structure containing XML and other data files. The benefits of this approach resemble those of the OOXML approach. One significant drawback is that, in some cases, it may not retain the original document formatting or content because of differing standards. Also, a 1-to-1 mapping of all fields may not exist. This type of change also can occur when converting to OOXML, but arguably, to a lesser extent. Schemas are being developed to enable OOXML conversion to ODF (and ODF to OOXML), so interoperability between these formats likely will be enabled as much as possible. Additionally, MS may offer native support for ODF in Office 12 Service Pack 2 (SP2) and Office 14. These Office versions will have native ability to write and read ODF files without requiring conversion by a separate application.

Native bit parsing requires pre-OOXML binary file structure knowledge and how each data stream portion describes the file elements. Traditionally, this knowledge was difficult to obtain, although some developers have obtained sufficient knowledge to use this approach. As part of the push to make the Office file formats more open, MS announced, via its Web site on March 27, 2007, that it may make the specifications for *certain MS Office binary file formats* available to developers whose needs are not met by OOXML or the other currently available approaches.^[2] MS subsequently published the first version of the documentation to its Web site.^[3] Although one can obtain pre-OOXML binary format details more easily, the format remains relatively complex. OOXML benefits make it a better option for development and for interoperability with other non-MS Office applications.

The historical difficulties in manipulating Office files (prior to OOXML use) led developers to pursue varied approaches, each with benefits and drawbacks. The nuances of each approach also affect how difficult or easy any particular type of inspection or sanitization technique is. This partially contributes to why existing tools have such varying feature sets and capabilities. OOXML enables developers to access all file components and more thoroughly process MS Office files against the set of minimum sanitization requirements that this document defines. Developers already have created applications to sanitize many of the areas identified here for pre-OOXML files. One example is Oracle's^{®5} Outside In Clean Content.^[4] This document will help

⁵ Oracle is a registered trademark of Oracle Corp.

identify minimum inspection and sanitization requirements for new tools that apply to the OOXML format as developers incorporate support for this newer format.

1.4 Document Organization

The document describes the elements in a MS 2007 document. It refers to these elements as constructs. In addition to describing each construct, this document describes the construct's potential flaws. For each file type (Word, Excel and PowerPoint), this document describes the construct's location, XML Schema Definition (XSD) of the syntax (called its grammar), construct processing recommendations, construct examples, and references. In some cases, detailed descriptions remain the same for all three Office products. For others, the document breaks out detailed descriptions by file type. Each construct descriptions begins with 'Office 2007.#:' and ends with 'Office 2007.#: END', where # represents a sequential index. Section 4 lists all of these constructs.

The following table summarizes the organization of this document.

Table 1-1. Document Organization

Section	Description
Section 1: Scope	This section describes the document's scope, background, organization, recommendation actions, and scope limitations.
Section 2: Constructs and Taxonomy	This section describes construct definitions and taxonomy for terms that appear in this document.
Section 3:: Office 2007 File Structure	This section describes the Office 2007 file structure.
Section 4: Office 2007 Constructs and Metadata	This section describes the constructs, data structures and metadata used in Office 2007 documents.
Section 5: XML Schema Constructs	This section details the XML for applicable constructs.
Section 6. Acronyms	This section lists the acronyms that appear in this document.
Section 7: Referenced Documents	This section lists the references that appear in this document.
Appendix A: BNF Grammar	This appendix contains the BNF grammar for the constructs in this document.

1.5 Recommendations

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

NOTE



The recommendations in this document are brief explanations rather than a How-To-Guide. Refer to the construct description or Microsoft'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.
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 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 type of data and pass the data onto an external action for handling that type of data (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 in this ISG 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 companion document to this, the *Data Transfer Guidance (DTG) for Microsoft Office 2007*, specifies four options for each recommended action: *Mandatory*, *Recommended*, *Optional*, or *Ignore*. Depending on the circumstances (e.g., a low to high data transfer versus a classified to unclassified transfer), programs can be configured to handle constructs different.

Table 1-3 summarizes the *Recommendation* action options.

Table 1-3. Recommendation Action Options

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

Each format documented for inspection and sanitization programs in this *ISG* appears in the companion *DTG* document. The *DTG* serves as a checklist for administrators and others to describe expected behaviors for inspection and sanitization programs. For example, 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 the *ISG* and enumerated recommendations in a row. After the recommendations, the worksheet displays a cell for each possible destination domain. This enables administrators 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 one 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 risk type; 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 level of risk level for that construct (i.e., high, medium, and low). This enables administrators to assign the risk type and risk level to each specific construct.

1.7 Document Limitations

The following subsections describe the scope limitations of this document, including those pertaining to constructs, covert channels, and character encoding.

1.7.1 Constructs

Although this document delves into many low level constructs in Office 2007 documents, it does not serve as a complete reference to all of the different constructs in the Office 2007 formats. The document covers the overall file format, product-specific constructs that govern the specific file formats (e.g., .docx, .xlsx, .pptx, .docm, .xlsm and .pptm) and metadata elements. We identified these as particular areas of concern for developers of file inspection and sanitization programs.

Recommendations appear within each of the metadata constructs. For the purposes of this document, these recommendations are “alternatives”. So, some recommendations may seem better than others and some recommendations may be more difficult to implement. Some recommendations complement each other and can be grouped together (e.g., “Remove headers” and “Remove footers”). Other recommendations may seem contradictory (e.g., “Remove Comments” and “Replace Comments”).

Readers may also note a lack of recommendations for simply logging the occurrences of metadata structures. Readers should understand that all activity of a file inspection application should be logged “at an appropriate level”. This touches on the topic of manual human review. People can interpret data in ways that computers cannot, especially regarding sound and images. For example, automated tools may pass a Joint Photographic Experts Group (JPEG) image of a sensitive document. As a result, human review can provide a valuable service in sanitizing documents, including documents that underwent automated inspection.

Often, the metadata construct will detail some underlying data structures of interest. The reader should not consider this document a “how-to guide”. The locations, recommendations, and examples do not reflect every avenue for identifying the metadata. The shift to OOXML offers improvements in Office file understandability but

OOXML is still a complex structure. Therefore, OOXML elements can have dependencies that are not immediately apparent, and removal or alteration of these elements could have a “ripple effect” on other data structures within OOXML.

This document contains references to specific Office XML files and their paths. The OOXML format offers flexibility in the location and naming of its internal XML files. However, it becomes cumbersome to refer explicitly to this feature each time this document references OOXML files and paths (e.g., “scan the main document as indicated by the path and filename in the [Content_Types].xml file” versus “scan the document* xml file”). The reader should understand that the default paths and filenames do not apply in every OOXML file.

Many applications can read from and save to Office 2007 file formats. When these applications save, using the Office 2007 file format, the underlying format of the OOXML file may differ (including Office 2003 saving documents as Office 2007). This document does not examine these variations.

In comparing this document with similar Office 2003 documents, as a result of feature changes between Office 2003 and 2007, some Office 2003 attributes do not appear in Office 2007. Examples include the following attributes:

- Meeting Minder
- Routing Slip
- Versions
- Fast Save Data
- Compound File Binary (CFB) Headers
- MS Globally Unique Identifier (GUID)
- Presentation Object Header
- Standard Presentation Object
- MetaFile Presentation Object
- Bitmap Presentation Object
- Device-Independent Bitmap (DIB) Presentation Object
- Generic Presentation Object
- Compound Object Header
- Compound Object Stream
- File Information Block (FIB)
- Future Records

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 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, this document does not discuss steganography within block text or media files, such as a hidden message that is embedded in 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

Characters used in words and sentences are grouped into a character set. A numeric representation, or code point that can be 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.

With XML, at the core of the new OOXML standard, the XML 1.0 specification determines permitted character encodings. The XML 1.0 specification dictates that no other character encodings beside Unicode Transformation Format-8 (UTF-8) or UTF-16 are permitted. In addition, consumers and producers of OOXML-compliant documents must enforce this requirement on embedded XML parts of a package.

The UTF-8 and UTF-16 character encodings are part of the Unicode standard, which supports bidirectional text, known as Right-to-Left (RTL) and Left-to-Right (LTR) text. Common scripts that use RTL include Hebrew, Persian, and Arabic. Bidirectional text properties for Word documents can be found in the package's /word/styles.xml part with the w:lang element and w:bidir attribute and also in the /word/document.xml with the w:rtl element. The underlying XML file stores RTL text as LTR, and the producing application renders the bidirectional text correctly.

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 these glyphs lose their original key mappings after saving in a document, they still pose a threat by concealing obfuscated text.

The use of embedded fonts may also pose issues for file inspection and sanitization programs. Embedded fonts are used to store fonts that may not be available on a target

computer. Embedded fonts have hidden malicious code that exploits flaws in Web browsers and could also hide data. In OOXML documents, embedded fonts are commonly stored in a “fonts” subdirectory.

2. CONSTRUCT AND TAXONOMY OVERVIEW

2.1 Constructs

This document attempts to formalize data structure definitions and the potential risks of these data structures by describing each element (or “constructs”) 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 posed by the element. For example, some metadata elements can cause inadvertent data leakage and others can be used for data exfiltration.
- **Location:** provides a textual description of where to find the element in the document. This can vary by client (or product) or it may apply across the entire product line.
- **Examples:** if applicable, the definition will contain an example of the construct.
- **Recommendations:** as described in Section 2.2.

This document formally defines the construct in a grammar (see Appendix A). The formalization attempts to bring a level of precision to the construct’s definition and serve as guidance for developers who write code to parse and handle the data structure. However, we strongly discourage developers from blindly applying the grammar. Rather, they should use standard software engineering practices and vigorous testing to ensure program correctness.

The start of each construct is denoted via a numbering schema that adheres to the syntax: **OFFICE 2003.x.y: Title**, where “x” represents the chapter in this document, “y” 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.y: END**.

2.2 Taxonomy

Table 2-1 describes the terms that appear in this document.

Table 2-1 Document Taxonomy

Term	Definition
Consistency	A construct state in which its 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 2007 format
DTG	A list of all ISG constructs and their associated recommendations. DTGs are 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 malicious data or code transmission
ISG	A document (such as this) that details a file format or protocol and inspection and sanitization activities for constructs within that file format
Recommendations	A series of actions for handling a construct when performing inspection and sanitization activities
Referential Integrity	The construct state in which all associated objects are properly referenced in the construct and that construct entries reference existing objects

3. OFFICE 2007 FILE STRUCTURE OVERVIEW

3.1 OFFICE 2007 OOXML

OOXML is a format that Microsoft developed to store Word, Excel, and PowerPoint files (and their subtypes, such as drawings). Microsoft places a combination of XML files and other file types (including images, other packages, and binary content) in a zip container that complies with the Open Packaging Convention (OPC). A user can unzip the OPC-compliant package and view the contents using a standard zip application, such as the one included with Windows XP Professional.

The container structures for the basic types of Word, Excel, and PowerPoint are similar. After unzipping the file container, typically three folders exist. (The number of subfolders may vary depending upon the file's content types.) The three main folders are the *docProps*, *_rels*, and the *word*, *xl*, or *ppt* folder (depending upon the used application). Note that the component names used here merely are conventions. The OPC does not specify the actual names except *[ContentTypes].xml*. (See the OPC documentation for details.) One should not hard code the names into software applications.

The *docProps* folder contains the application properties (*app.xml*), core properties (*core.xml*), and possibly custom properties (*custom.xml*). One can also configure each application to insert a thumbnail of the first page of the file into this folder. (PowerPoint creates a thumbnail by default.) The *_rels* folder contains schema references for the files in the *docProps* folder and in the *word*, *xl*, or *ppt* folder.

The *[Content_Types].xml* file lists all of the content types within the document, including references to the appropriate schema. If a content type exists within the package that is not defined in the content types and the *rels* files, the Office application may fail to open the document. The *[Content_Types].xml* contains a reference to items, such as the binary object schema for OLE Objects and the Enhanced Metafile (EMF) schema for the image that represents the content. An example *[Content_Types].xml* file appears later in this document.

The application-specific folder (*word*, *xl*, or *ppt*) contains subfolders to store the different file content types. Media, such as images, reside in the media folder. Embeddings generally reside in the embeddings folder, although some binary objects such as macros may reside in the application-specific folder instead. Theme information resides in the theme folder. Charts and drawings reside in their respective folders.

The main Word file is `word/document.xml`. Figure 3-1 shows an example structure of a simple Word 2007 document. This, and subsequent Excel and PowerPoint examples, do not exhaust all possible files and folders because the files could contain charts, drawings, headers, footers, and other content types (which do not appear for simplicity reasons).

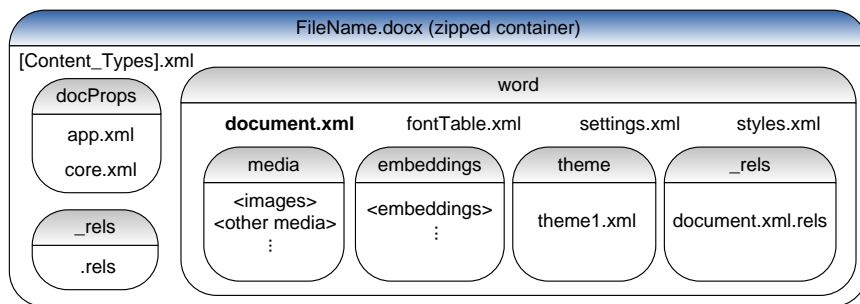


Figure 3-1 Word Format

The main Excel file is `xl/workbook.xml`. Sheets within the workbook reside in `xl/worksheets`, with the text information stored in `sharedStrings.xml`. Text may also reside in-line within the main file. Figure 3-2 shows an example of this layout structure.

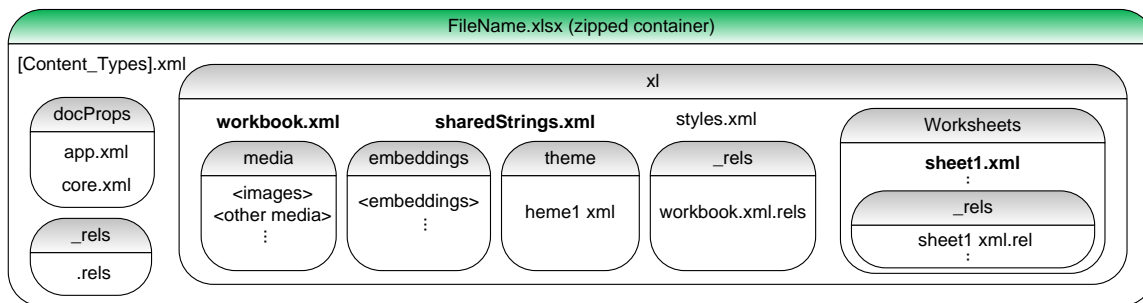


Figure 3-2: Excel Format

The main PowerPoint file is `ppt/presentation.xml`. The content slides reside in the *Slides* folder. Layout and master information resides in separate subfolders for each content type. Figure 3-3 shows an example of the layout.

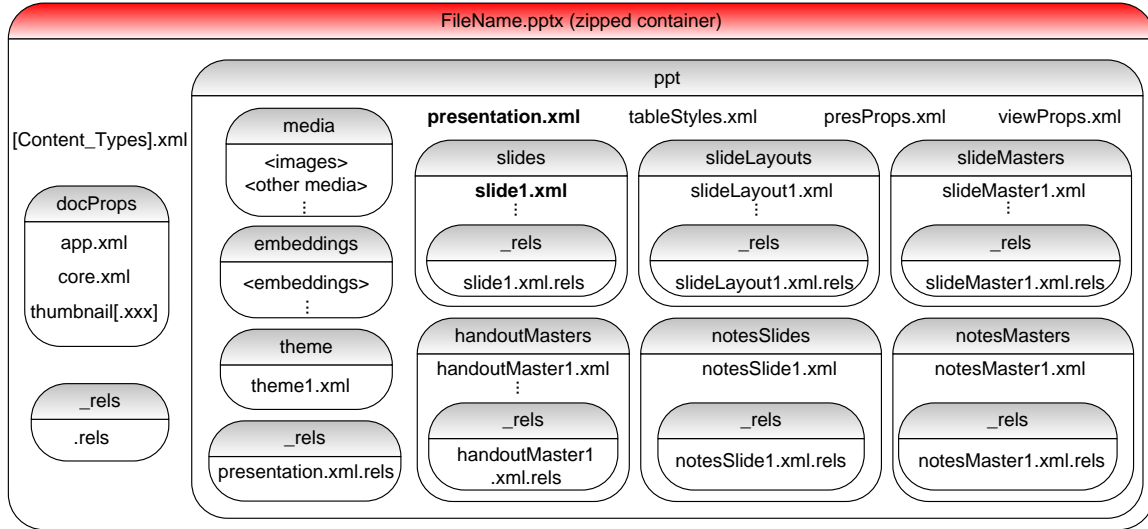


Figure 3-3: PowerPoint Format

3.2 Embedded and Linked Objects

An OOXML file may contain a variety of embedded content types, including more OOXML files (which may also contain more embedded files). A significant consideration for inspection and sanitization is the recursion level that any utility provides. One can think of recursion, the ability to inspect and sanitize files contained in a file, in terms of depth and width. For example, if a Word file contains another Word file, the inspection would need to include recursion one level deep. If the embedded Word file also contains an embedded Word file, then the recursion capabilities would need to go two levels deep. Recursion width also can present an issue. For example, an Office file, such as a Word document, may contain two un-nested Word files. This example would result in a recursion depth of one and a width of two. Recursion can impose a significant performance penalty on inspection and sanitization tools, especially as depth and width of recursion increases. Recursive processing potentially may generate complex analysis results also, which may not be intuitive for average users to understand.

Two perspectives for processing include inspection and sanitization. For inspection, the tool should attempt to identify all embedded items. If the embedded items are OOXML files, the tool may also attempt to perform a full inspection on those files. If the tool supports inspecting embedded OOXML files, the tool should allow configuration to select how wide and deep the tool should perform in-depth inspection. Minimally, the

tool should recognize the presence of embeddings, their types, and their sizes (if feasible). For sanitization, the need for recursive processing may vary between environments, based upon the risk level.

Developers may configure, but should limit, recursion width in relation to the sanitization application's performance capabilities. Developers can establish the limit as a configurable setting, but they should set reasonable default limits based on the application's resource intensiveness.

When a user copies an Office object from one window to another, the source file may copy into the destination file without informing the user. In many cases, an additional alternative representation inserts into the document. In most cases, this is an image, such as an EMF or Windows MetaFile (WMF). In some cases, this may be a table or other representation. Users only see the representative image and may not know that an embedded object exists until they click the object. For sanitization, the ability to retain the alternative representation enables a tool to remove the embedded object while retaining document's appearance.

A significant potential for unknowingly embedding content in an Office file exists. Often, the embeddings reside as binary content. One of the most common ways that binary content is embedded in OOXML files is when a pre-OOXML file is converted to OOXML. Most embeddings and some types of content (e.g., macros) likely will be saved as binary content inside the resulting OOXML file.

Other scenarios exist for a user to unintentionally introduce binary content into an OOXML file. For example, a user might copy a Visio diagram and paste it into a Word document. Unless the user chose the 'Paste Special' option from Word's Edit menu and selected an image type (e.g., EMF), the Visio document will likely embed within the file as pre-OOXML binary object. In addition to the binary embedding, an EMF representation will be created and display within the document. Figure 3-4 shows a simplified overview of the Word document structure.

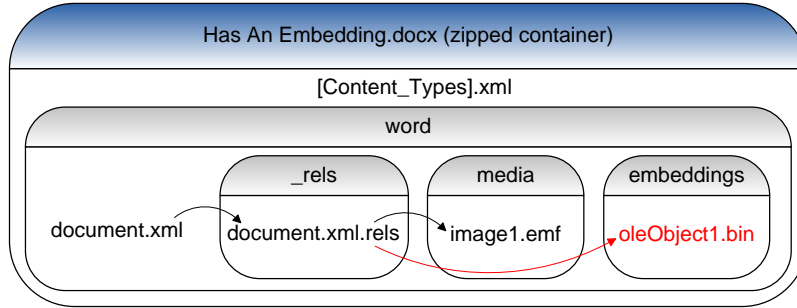


Figure 3-4: Embedded Content

The content is described in the document through a series of relationships. At a high level, the *[Content_Types].xml* file contains references to the object types that exist within the file. In the example shown here, the file contains an OLE object (a pre-OOXML data type) and an EMF.

File: [Content_Types].xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<Types xmlns="http://schemas.openxmlformats.org/...">
  <Default Extension="bin" ContentType="application/vnd.openxmlformats-
officedocument.oleObject" />
  <Default Extension="emf" ContentType="image/x-emf" />
  :
  :
</Types>
```

The data types that were identified within the *[Content_Types].xml* are referenced in the document where they are used. A reference resides in *document.xml* to the *document.xml.rels* file with relationship identifiers where the object components exist. The references appear in the example below where “rId4” and “rId5” are highlighted.

File: word\document.xml

```

<w:object w:dxaOrig="9181" w:dyaOrig="11881">
:
:
  <v:shape id="_x0000_i1025" type="#_x0000_t75"
style="width:459pt;height:594pt" o:ole="">
    <v:imagedata r:id="rId4" o:title="" />
  </v:shape>
  <o:OLEObject Type="Embed" ProgID="Visio.Drawing.11"
ShapeID="_x0000_i1025" DrawAspect="Content" ObjectID="_1246946141"
r:id="rId5" />
</w:object>

```

The *document.xml.rels* file points to the folders where the content resides, along with the schema reference for the content type. This is where the references from the document.xml file become associated with specific target objects.

File: word_rels\document.xml.rels

```

<Relationships xmlns="http://schemas.openxmlformats.org/...">
:
:
  <Relationship Id="rId5"
Type="http://schemas.openxmlformats.org/officeDocument/2006/relations/oleObject" Target="embeddings/oleObject1.bin" />
  <Relationship Id="rId4"
Type="http://schemas.openxmlformats.org/officeDocument/2006/relations/image" Target="media/image1.emf" />
</Relationships>

```

The logical structure for storing embedded objects enables developers to remove embeddings effectively while retaining the associated representation to maintain the look of the Office file after sanitization.

3.3 Office 2007 Zip File Structure and Format

As stated earlier, Office 2007 documents are zip files that contain various files and folders. The zip file format stores the components of the OOXML document in a single archive. Although documented in detail in other sources (see the references for each construct), this file format discussion focuses on the inspection of the areas of concern. The intent is to ensure that inspection and sanitization programs consider zip file data structures. However, one should view the references as canonical.

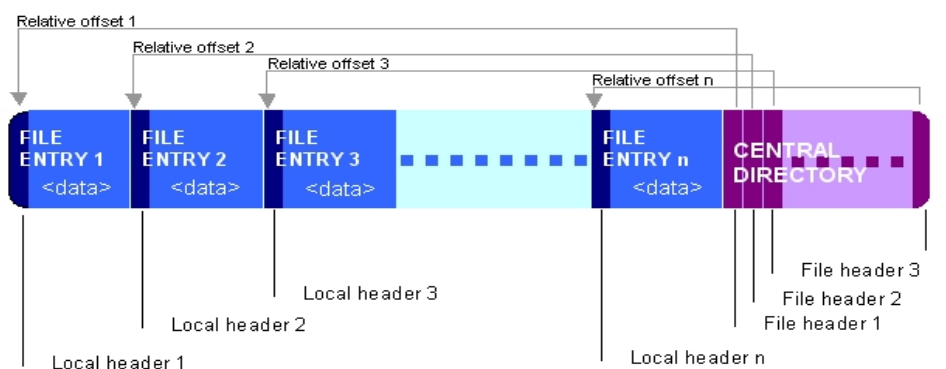
The zip file contains files and directories, each stored in arbitrary order. Files are stored first, followed by data structures representing the directory layout. The order of the file

entries in the directory does not have to conform to the order of file entries in the archive--the format is non-sequential.

The central directory indicates the file location. It appears at the end of the zip file. File entries represent each of the files and directories.

A local header introduces each file entry and contains file information, such as the compression method, zip file version, and file size. The file name immediately follows the local file header, followed by an optional data description (used for streaming applications), and then actual file data (perhaps encrypted and compressed). The local file header contains a 16 bit-field called the extra data field. It represents the "extra" data that is appended to the end of the file data. The "Extra" data fields are the key to zip format extensibility. The "Extra" fields are exploited to support ZIP64 formats, WinZip-compatible Advanced Encryption Standard (AES) encryption, and New Technology File System (NTFS) file timestamps. In theory, many other extensions are possible via this coded "extra" field. Client applications that do not recognize the content contained in the extra data field must ignore and persist.

The central directory consists of file headers holding, among other information, the file names and the relative offset in the archive of the local headers for each file entry. Each file entry is marked by a specific four-byte "signature". Each central directory entry is marked also with a different particular four-byte signature. Zip file parsers typically look for the appropriate signatures when parsing a zip file. No beginning of file (BOF) or end of file (EOF) markers in the zip specification exists. Instead, zip tools scan for the signatures of the various fields.



[http://en.wikipedia.org/wiki/ZIP_\(file_format\)](http://en.wikipedia.org/wiki/ZIP_(file_format))

Figure 3-5 Zip File Format

Overall Zip File Format:

[local file header 1]
 [file data 1]
 [data descriptor 1]
 ...
 [local file header n]
 [file data n]
 [central directory]
 [zip64 end of central directory record]
 [zip64 end of central directory locator]
 [end of central directory record]

OFFICE 2007.3.1: Local File Header 1

DESCRIPTION:

The local file header contains 13 components with information about the file, such as the file name, signature, version needed to extract, general purpose bit flag, compression method, last modified time and date, Cyclic Redundancy Check -32 (CRC-32), compressed and uncompressed size, extra field, and file name and extra field lengths.

The "extra" data fields support the extensibility of the zip format. "Extra" fields are exploited to support ZIP64 formats, WinZip-compatible AES encryption, and NTFS file timestamps. However, there are many other extensions possible via this coded "extra" field that may cause a concern.

A specific 4-byte "signature" marks each file entry. Likewise, each entry in the central directory is marked with a different particular 4-byte signature. Zip file parsers typically look for the appropriate signatures when parsing a zip file.

A CRC is a non-secure hash function that is designed to detect accidental changes to raw computer data. CRC-32 is popular and is Institute of Electrical and Electronics (IEEE) recommended.

	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	0xe	0xf		
0x0000	Signature				Version		Flags		Compression		Mod:time		Mod: date		Crc-32			
0x0010	Crc-32		Compressed size				Uncompressed size				File name len		Extra field len					
0x0020									File name (variable size)									
0x0030									Extra field (variable size)									

<https://users.cs.jmu.edu/buchhofp/forensics/formats/pkzip.html>

CONCERN:

The extra field is an area that can be easily store up to 2^{16} bytes of unauthorized content.

PRODUCT: WORD, EXCEL, POWERPOINT

LOCATION:

The location of the local file header is at the beginning of the file. Its first four bytes contain the local file header signature. It ends with its variable length file name and extra field. Subsequent file entries appear after the previous file's file data and any "extra" data.

RECOMMENDATION:

AR.1 Validate: Ensure consistency by verifying that the calculated CRC-32 matches the CRC-32 in the local file header.

AR.2 Remove: Remove the extra data field and set the extra data field length member to 0.

AR.3 Replace: N/A

AR.4 External Filtering Required: Scan the file name field with an application that is configured for filtering text data.

AR.5 Review: N/A

EXAMPLE:

A sample local file header:

```
00000000 50 4b 03 04 14 00 00 00 08 00 1c 7d 4b 35 a6 e1 |PK.....}K5..|
00000010 90 7d 45 00 00 00 4a 00 00 00 05 00 15 00 66 69 |.E...J.....fi|
00000020 6c 65 31 55 54 09 00 03 c7 48 2d 45 c7 48 2d 45 |le1UT...H-E.H-E|
00000030 55 78 04 00 f5 01 f5 01 0b c9 c8 2c 56 00 a2 92 |Ux.....,V...|
```

This results in the following fields and field values:

	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	0xe	0xf
0x0000	50	4b	03	04	14	00	00	00	08	00	1c	7d	4b	35	a6	e1
0x0010	90	7d	45	00	00	00	4a	00	00	00	05	00	15	00	66	69
0x0020	6c	65	31	55	54	09	00	03	c7	48	2d	45	c7	48	2d	45
0x0030	55	78	04	00	f5	01	f5	01								

REFERENCE:

<http://www.pkware.com/documents/casestudies/APPNOTE.TXT>

<https://users.cs.jmu.edu/buchhofp/forensics/formats/pkzip.html>

http://www.enterag.ch/enterag/downloads/Zip64File_TechnicalDocumentation.pdf

Brayer, K; Hammond, J L Jr. (December 1975). "Evaluation of error detection polynomial performance on the AUTOVON channel". *National Telecommunications Conference, New Orleans, LA*. Conference Record. 1. New York: Institute of Electrical and Electronics Engineers. pp. 8-21 to 8-25.

OFFICE 2007.3.1: **END**

OFFICE 2007.3.2: **File Data 1**

DESCRIPTION:

The file data is the component of the zip archive that stores compressed or uncompressed data for the file.

CONCERN:

Arbitrary data may be added, removed, or altered without authorization.

PRODUCT: WORD, EXCEL, POWERPOINT

LOCATION:

The file data immediately follows the local file header. The last two components of the local file header are its variable-sized file name and extra length. Thus, the byte immediately after the last byte of the extra length starts the file data.

RECOMMENDATION:

AR.1 Validate: Ensure consistency by decompressing and, if necessary, decrypting the file data. If either process fails, the file data is invalid. Additionally, calculate the CRC-32 of the file data and compare it to the local file header value.

AR.2 Remove: N/A

AR.3 Replace: N/A

AR.4 External Filtering Required: N/A

AR.5 Review: N/A

REFERENCE:

<http://www.pkware.com/documents/casestudies/APPNOTE.TXT>

OFFICE 2007.3.2: **END**

OFFICE 2007.3.3: **Central Directory Structure**

DESCRIPTION:

The Central Directory consists of file headers holding, among other information, the file names, and the relative offset in the archive of the local headers for each file entry.

The Central Directory structure may store compressed and encrypted data. Although not required, the assumption when encrypting the Central Directory structure is that it will be compressed for greater storage efficiency. The section that describes the Strong Encryption Specification also contains information on the Central Directory encryption feature. The Digital Signature record will not be compressed or encrypted.

Structure

```
[file header 1]
.
.
.
[file header n]
[digital signature]
```

File Header Structure

<https://users.cs.jmu.edu/buchhofp/forensics/formats/pkzip.html>
<https://users.cs.jmu.edu/buchhofp/forensics/formats/pkzip.html>

	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	0xe	0xf
0x0000	Signature				Version		Vers. needed		Flags		Compression		Mod. time		Mod. date	
0x0010	Crc-32				Compressed size				Uncompressed size				File name len		Extra field len	
0x0020	File comm. len		Disk # start		Internal attr.		External attr.				Offset of local header					
0x0030	File name (variable)															
0x0040	Extra field (variable)															
0x0050	File comment (variable)															

<https://users.cs.jmu.edu/buchhofp/forensics/formats/pkzip.html>

CONCERN:

Arbitrary data can be added into the directory. For example, the extra field may hide unauthorized code or data. Also, if the Central Directory structure is corrupted, the entire file may lock-up and cause a denial-of-service attack.

PRODUCT: WORD, EXCEL, POWERPOINT

LOCATION:

The Central Directory could appear right after the file data if the data descriptor, archive decryption header, and archive extra data record are not present. However, if any of these reside in the zip file, the Central Directory will appear after the components. Compare the first four bytes after the data file to the Central Directory structure's signature. If the four bytes match, the start of the Central Directory has been located.

RECOMMENDATION:

AR.1 Validate: Ensure referential integrity by comparing the components listed in the directory with the actual file contents.

AR.2 Remove: Remove any extra field data and set the extra field data length to 0. Additionally, remove comments and set the comment length to 0.

AR.3 Replace: N/A

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

AR.5 Review: N/A

EXAMPLE:

The corresponding file header from the local file header example above starts at byte 0x9a2 in the example file:

```
000009a0 28 f0 50 4b 01 02 17 03 14 00 00 00 08 00 1c 7d |(.PK.....)|
000009b0 4b 35 a6 e1 90 7d 45 00 00 00 4a 00 00 00 05 00 |K5...)E...J....|
000009c0 0d 00 1c 00 00 00 01 00 00 00 a4 81 00 00 00 00 |.....|
000009d0 66 69 6c 65 31 55 54 05 00 03 c7 48 2d 45 55 78 |file1UT...H-EUx|
000009e0 00 00 74 68 69 73 20 69 73 20 61 20 63 6f 6d 6d |..this is a comm|
000009f0 65 6e 74 20 66 6f 72 20 66 69 6c 65 20 31 50 4b |ent for file 1PK|
```

	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	0xe	0xf
0x0000	50	4b	01	02	17	03	14	00	00	00	08	00	1c	7d	4b	35
0x0010	a6	e1	90	7d	45	00	00	00	4a	00	00	00	05	00	0d	00
0x0020	1c	00	00	00	01	00	00	00	a4	81	00	00	00	00	66	69
0x0030	6c	65	31	55	54	05	00	03	c7	48	2d	45	55	78	00	00
0x0040	74	68	69	73	20	69	73	20	61	20	63	6f	6d	6d	65	6e
0x0050	74	20	66	6f	72	20	66	69	6c	65	20	31				

<https://users.cs.jmu.edu/buchhofp/forensics/formats/pkzip.html>

REFERENCE:

<http://www.pkware.com/documents/casestudies/APPNOTE.TXT>

<https://users.cs.jmu.edu/buchhofp/forensics/formats/pkzip.html>

<http://www.binaryessence.com/dct/en000221.htm>

OFFICE 2007.3.3: **END**

OFFICE 2007.3.4: **Zip64 End of Central Directory Record**
DESCRIPTION:

The ZIP64 End of Central Directory record indicates the end of the Central Directory structure for Zip64 files, providing information about the Central Directory location (e.g., offset address, file or disk identification containing the directory, size of the directory, and number of files).

The value stored into the "size of zip64 end of central directory record" should reflect the size of the remaining record and should not include the leading 12 bytes.

Size = SizeOfFixedFields + SizeOfVariableData - 12.

Special purpose data may reside in the zip64 extensible data sector field following either a V1 or V2 version of this record. To ensure identification of this special purpose data it must include an identifying header block consisting of the following information:

Header ID 2 bytes
Data Size 4 bytes

The *Header ID* field indicates the type of data that resides in the data block that follows. *Data Size* identifies the number of bytes that follow for this data block type. Multiple special purpose data blocks may exist, but a *Header ID* and *Data Size* field must precede each data block.

CONCERN:

If the ZIP64 Central Directory record is missing or incomplete, the client software to open the zip file will have an error.

PRODUCT: WORD, EXCEL, POWERPOINT

LOCATION:

The ZIP64 central directory could appear right after the file data if the data descriptor, archive decryption header, and archive extra data record are not present. However, if any of them appear in the zip file, the Central Directory will follow the components. Compare the first four bytes after the data file to the Central Directory structure's signature. If both four bytes match, the start of the Central Directory has been located.

RECOMMENDATION:

AR.1 Validate: Ensure the referential integrity of the Central Directory structure and the file components.

AR.2 Remove: Remove the extensible data sector and set length parameters to 0.

AR.3 Replace: N/A

AR.4 External Filtering Required: Identify the data type in the extensible data field and pass it to the action that is configured for that data type.

AR.5 Review: N/A

REFERENCE:

<http://www.pkware.com/documents/casestudies/APPNOTE.TXT>

<http://www.binaryessence.com/dct/en000221.htm>

<https://users.cs.jmu.edu/buchhofp/forensics/formats/pkzip-images/end-of-central-directory-record.png>

OFFICE 2007.3.4: **END**

OFFICE 2007.3.5: **Zip64 End of Central Directory Locator****DESCRIPTION:**

Similar to the Zip64 End of Central Directory record, this also signifies the termination of the Central Directory in ZIP64™ format. However, the Zip64 End of Central Directory locator provides less information than the *Zip64 End of Central Directory* Record. It only contains its signature, the number of disks from the start of the Zip64 end of Central Directory, its relative offset, and the total number of disks.

CONCERN:

If the locator is missing or incomplete, the client software, used to open the zip file, will have an error, and will not open the file.

PRODUCT: WORD, EXCEL, POWERPOINT**LOCATION:**

The Zip64 end of Central Directory locator is located between the *Zip64 end of Central Directory* record and the end of *Central Directory* record.

RECOMMENDATION:

AR.1 Validate: Ensure consistency by verifying that an end of Central Directory structure is present in the file.

AR.2 Remove: Remove all data after this structure.

AR.3 Replace: N/A

AR.4 External Filtering Required: Identify any data after the structure and pass it to the action that is configured for that data type.

AR.5 Review: N/A

REFERENCE:

<http://www.pkware.com/documents/casestudies/APPNOTE.TXT>

<http://www.binaryessence.com/dct/en000221.htm>

OFFICE 2007.3.5: **END**OFFICE 2007.3.6: **End of Central Directory Record****DESCRIPTION:**

This signifies the termination of the Central Directory. It provides information about the Central Directory location (e.g., offset address, identification of file or disk containing the directory, size of the directory, and number of files).

The structure of the *end of Central Directory* record is as follows:

	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	0xe	0xf
0x0000	Signature			Disk number		Disk # w/cd		Disk entries		Total entries		Central directory size				
0x0010	Offset of cd wrt starting disk				Comment len		ZIP file comment (variable)									

<https://users.cs.jmu.edu/buchhofp/forensics/formats/pkzip.html>

Signature	The signature of <i>end of Central Directory</i> record. This is always '\x50\x4b\x05\x06'.
Disk Number	The number of this disk (containing the <i>end of Central Directory</i> record)
Disk # w/cd	Number of the disk on which the Central Directory starts
Disk entries	The number of Central Directory entries on this disk
Total entries	Total number of entries in the Central Directory
Central directory size	Size of the Central Directory in bytes
Offset of cd wrt to starting disk	Offset of the start of the Central Directory on the disk on which the Central Directory starts
Comment len	The length of the following <i>Comment</i> field
ZIP file comment	Optional comment for the zip file

CONCERN:

If the end of the Central Directory record is missing or incomplete, the client software to open the zip file will have an error and will not open the file.

PRODUCT: WORD, EXCEL, POWERPOINT

LOCATION:

The *end of Central Directory* record is located right after the *zip64 end of Central Directory* locator. It is the last element of the zip file format.

RECOMMENDATION:

AR.1 Validate: Ensure consistency by verifying that an end of Central Directory structure is present in the file.

AR.2 Remove: Remove all data after this structure.

AR.3 Replace: N/A

AR.4 External Filtering Required: Identify any data after the structure and pass it to the action that is configured for that data type.

AR.5 Review: N/A

EXAMPLE:

The end of Central Directory in the example file starts at byte 0xb36:

00000b30	6f	6d	6d	65	6e	74	50	4b	05	06	00	00	00	00	04	00	ommentPK.....
00000b40	04	00	94	01	00	00	a2	09	00	00	33	00	74	68	69	733.this
00000b50	20	69	73	20	61	0d	0a	6d	75	6c	74	69	6c	69	6e	65	is a..multiline
00000b60	20	63	6f	6d	6d	65	6e	74	20	66	6f	72	20	74	68	65	comment for the
00000b70	20	65	6e	74	69	72	65	20	61	72	63	68	69	76	65		entire archive

	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	0xe	0xf
0x0000	50	4b	05	06	00	00	00	00	04	00	04	00	94	01	00	00
0x0010	a2	09	00	00	33	00	74	68	69	73	20	69	73	20	61	0d
0x0020	0a	6d	75	6c	74	69	6c	69	6e	65	20	63	6f	6d	6d	65
0x0030	6e	74	20	66	6f	72	20	74	68	65	20	65	6e	74	69	72
0x0040	65	20	61	72	63	68	69	76	65							

<https://users.cs.jmu.edu/buchhofp/forensics/formats/pkzip.html>

Signature	'\x50\x4b\x05\x06'.
Disk Number	0
Disk # w/cd	0
Disk entries	4
Total entries	4
Central Directory size	0x194 = 404 bytes
Offset of cd wrt to starting disk	byte 0x9a2 = byte 2466
Comment len	0x33 = 51 bytes
ZIP file co	"this is a multiline comment for the entire archive"

mment

REFERENCE:

<http://www.pkware.com/documents/casestudies/APPNOTE.TXT>

<http://www.binaryessence.com/dct/en000221.htm>

<https://users.cs.jmu.edu/buchhofp/forensics/formats/pkzip.html>

OFFICE 2007.3.6: **END**

3.4 Macros and Malicious Content Overview

In addition to the risk of unintentional release of sensitive data, macros and malicious content may propagate through the Office file use. A new feature in Office 2007 enables users to create macro-enabled files that end in '.***m', and non-macro files in '.***x', where the *** is a doc, xls, or ppt file. Ostensibly, this provides an additional layer of security because it is easier to identify when a file may contain a macro; however, some potential ways to subvert this security exist. If one renames a '.docm' file to '.docx', Office will not open it (unless the schemas and content types are also updated). However, if one simply renames a '.docm' to a '.doc', Office generally opens it without warning about an incorrect file extension. Furthermore, if a '.docx' file contains another embedded Office file (such as a Word document that contains a macro-enabled Excel spreadsheet), macros and other malicious content still have the potential to execute. The binary Excel format, .xlsb, is also macro-enabled (but an exception to the rule that macro-enabled files have the form .***m). For these reasons, one should never rely upon the extension alone for file type and version identification.

Macro and executable risks can be significant. Past analysis within NSA's Systems and Network Analysis Center (SNAC) identified the risk level and types created by the presence of these features in Office documents.^[5,6,7] One may identify and remove macros and executable content from Office files without having to open the file in the authoring application. Office's security settings traditionally were the last line of defense against macros, as users had no way of telling if the file contained macros until they opened it in Office and received a warning (provided appropriate security setting establishment) that the file contained this content. Risk mitigation becomes more effective if one can remove malicious content before reaching the Office application.

Microsoft also provides a toolset called the MS Office Isolated Conversion Environment (MOICE), which allows users to open pre-OOXML files and convert them to OOXML, while removing many of the potential active content threats.^[8] NSA's SNAC has a fact

sheet for MOICE configuration, and Microsoft provides a deployment guide with MOICE.^[9] The features and configuration capabilities vary for each version of Office, so it is important to review these documents before relying on MOICE to reduce the threat of active content.

Twenty main extensions for the file types exist, as shown in Table 3-1. The most commonly used are '.docx', '.xlsx', and '.pptx'. The recommendations focus on those files and the macro-enabled variants '.docm', '.xlsm', '.pptm'.

Table 3-1: Office 2007 Extensions^[10]

Extension	Description
.docx, .xlsx, .pptx	Word, Excel, PowerPoint 2007 XML File
.docm, .xlsm, .pptm	Word, Excel, PowerPoint 2007 XML Macro-enabled File
.dotx, .xltx, .potx	Word, Excel, PowerPoint 2007 XML Template
.dotm, .xltm, .potm	Word, Excel, PowerPoint 2007 XML Macro-enabled Template
.xlsb	Excel 2007 (Non-XML) binary workbook
.xlam, .ppam	Excel, PowerPoint 2007 XML Macro-Enabled Add-in
.ppsx	PowerPoint 2007 XML Show
.ppsm	PowerPoint 2007 XML Macro-Enabled Show
.sldx	PowerPoint 2007 XML Slide
.sldm	PowerPoint 2007 XML Macro-Enabled Slide
.thmx	Office Theme

3.4.1 Font Embedding 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 problem source for the Office product suite and web applications (see, for example, this Microsoft [security bulletin](#).)

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

DESCRIPTION:

An embedded font structure holds the information to install a font temporarily so it can be rendered 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 structures can hold arbitrary, variable-sized data, one easily can use them to hide data and exploit software flaws within client applications. As noted, this data structure has been used for such activities.

PRODUCT: WORD**LOCATION:**

Word stores embedded fonts in the */word/fonts* directory within the package. The */word/fontTable.xml* file will also contain an XML construct similar to this for each embedded font:

```
<w:embedRegular w:fontKey='{FEADD543-344B-44FB-AEA9-E1AFDA698342}'
r:id='rId2' />
```

A line in *[ContentType].xml* similar to the following indicates that the embedded fonts are stored obfuscated, in which the first few bytes of an embedded font are stored scrambled.

```
<Default ContentType='application/vnd.openxmlformats-
officedocument.obfuscatedFont' Extension='odttf' />
```

PRODUCT: POWERPOINT**LOCATION:**

The presence of embedded fonts in an Office Open XML *PresentationML* document is identified by this XML construct in *[ContentTypes].xml*:

```
<Default Extension="fntdata" ContentType="application/x-
fontdata"/>
```

The actual font data reside under the *ppt/fonts* directory within the package.

RECOMMENDATION:

AR.1 Validate: Ensure consistency by validating the embedded font to make sure it meets the standard for embedded fonts.

AR.2 Remove: Remove all embedded fonts and the embedded font structures.

AR.3 Replace: Replace all bytes in the embedded fonts with a value to nullify the font.

AR.4 External Filtering Required: N/A

AR.5 Review: N/A

OFFICE 2007.3.7: **END**

4. OFFICE 2007 CONSTRUCTS AND METADATA

This section decomposes the MS Office 2007 constructs. For each construct, it provides an overview, and describes the concerns, applicable Office products, locations, examples (if applicable), recommendations and actions, and references.

OFFICE 2007.4.1: **Field Codes**

OVERVIEW:

Field codes automatically update certain information within 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—not to PowerPoint or Excel. To view Fields in MS Word, open the application and select the pull-down Insert > Quick Parts > Field. This displays the list of field items that can go into a Word document.

MS Word includes Fields in its “smart formatting” feature. Fields have a text representation in the document, and also have a definition, which gives the field the ability to update itself when users select the update field option.

CONCERNS:

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 of the document. This may reveal sensitive network architectures and could render document parts unreadable or missing after the document transfers outside of the domain of the linked file.

Category	Fields	Description & Concern
Date and Time	CREATEDATE	Inserts the current date and/or time, or date and/or time of some kind of event. Some dates are important to document functionality, but also may have the potential to leak information. CreateDate, in a legal document, can contain 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.
	DATE	
	EDITTIME	
	PRINTDATE	
	SAVEDATE	
	TIME	

	Document Automation	<p>COMPARE</p> <p>DOCVARIABLE</p> <p>GOTOBUTTON</p> <p>IF</p> <p>MACROBUTTON</p> <p>PRINT</p>	<p>Compares values and takes action based on outcome. Runs macros and sends a code to a printer</p> <p>If such automation is needed in the document, it is probably the main reason the document exists. Document automation is also associated with using Visual Basic⁶ in the document. Preserving document automation may also require leaving macros in the document. Preserving document automation is risky. Getting such documents through a filter may require establishing a special policy for only such files.</p>	
	Document Information	<p>AUTHOR</p> <p>COMMENTS</p> <p>DOCPROPERTY</p> <p>FILENAME</p> <p>FILESIZE</p> <p>INFO</p> <p>KEYWORDS</p> <p>LASTSAVEDBY</p> <p>NUMCHARS</p> <p>NUMPAGES</p> <p>NUMWORDS</p> <p>SUBJECT</p> <p>TEMPLATE</p>	<p>Inserts or stores information about the document</p> <p>A core set of basic fields provides support for nearly all MS Word documents. These fields are keys to preserving document edit ability and present little threat.</p>	

⁶ Visual Basic is a registered trademark of Microsoft Corp.

	TITLE	
Equations and Formulas	=formula ADVANCE EQ SYMBOL	<p>Defines formulas and calculates results. Inserts symbols</p> <p>Equations and formulas within a document could hide sensitive information while only displaying a computed value. However, removing these may break document functionality. One should determine a special policy on how to handle these fields appropriately.</p>
Index and Tables	INDEX RD TA TC TOA TOC XE	<p>Defines entries for, and builds a table of contents, table of figures, and tables of authorities</p> <p>These fields create indexes for Tables of Contents and items that are associated with setting and using bookmarks. They pose little threat.</p>
Links and References	AUTOTEXT AUTOTEXTLIST BIBLIOGRAPHY HYPERLINK INCLUDEPICTURE INCLUDETTEXT LINK NOTEREF PAGEREF QUOTE	<p>Inserts information from another place in the same document, from a different document or file, or from an AutoText entry.</p> <p>Fields such as INCLUDETEXT and INCLUDEPICTURE pull information from other files and display it. This applies to cases when a site has information that changes frequently, but needs to be published in several different documents. The information resides in one file and other documents use the Include Text field to pull in the most recent</p>

		REF STYLeref	version of the information. It is unlikely that both the documents are going 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 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	Specifies numbering for document items, such as sections, pages, and bar codes These fields are keys to preserving document edit abilities. They present little threat.

	SECTIONPAGES SEQ	
User Information	USERADDRESS USERINITIALS USERNAME	Stores or inserts the name, initials, or address of the document user
Form Field	FORMCHECKBOX FORMDROPDOWN FORMTEXT	Allows the insertion of form fields

PRODUCT: WORD

LOCATION:

The following description of the XML implementation of field codes from “Office Open XML Part 4: Markup Language Reference” summarizes the intent:

Fields shall be implemented in XML using either of two approaches:

As a *simple field implementation*, using the fldSimple element, or

As a *complex field implementation*, using a set of runs involving the fldChar and instrText elements.

For a simple field implementation, only one element, *fldSimple*, shall be used, in which case, its instr attribute shall contain a *field*, and the body of the element shall contain the most recently updated field result. [Example: Here is the corresponding XML for a simple field implementation of DATE:

```
<w:r>  
  <w:fldSimple w:instr="DATE"> 12/31/2005 </w:fldSimple>  
</w:r>
```

end example]

For a complex field implementation, a set of runs shall be used with each run containing, in sequence, the following elements:

fldChar with attribute fldCharType value begin,

One or more instrText elements, which, collectively, contain a complete *field*,

Optionally,

- fldChar with attribute fldCharType value separate, which separates the field from its field result,
- Any number of runs and paragraphs that contains the most recently updated field result, and

fldChar with attribute fldCharType value end.

[*Note:* Fields that are for display purposes only have no need to, and do not, store a field result. *end note*][*Example:* Here is the corresponding XML for a complex field implementation of DATE:

```
<w:r>
  <w:fldChar w:fldCharType="begin" />
</w:r>
<w:r>
  <w:instrText xml:space="preserve"> DATE </w:instrText>
</w:r>
<w:r>
  <w:fldChar w:fldCharType="separate" />
</w:r>
<w:r>
  <w:t>12/31/2005</w:t>
</w:r>
<w:r>
  <w:fldChar w:fldCharType="end" />
</w:r>
```

end example]

"Office Open XML Part 4: Markup Language Reference," §2.16.2

RECOMMENDATIONS:

WR.1 Validate: N/A

WR.2 Remove: Remove the field code definition and retain the generated text.

WR.3 Replace: Replace the generated text with provided text and retain the field code.

WR.4 External Filtering Required: N/A

WR.5 Review: N/A

EXAMPLE:

```
<w:p w:rsidR="00FC4ED8" w:rsidRDefault="00202BA1">
  <w:fldSimple w:instr="AUTHOR &quot;Thomas Jefferson&quot;; \* Upper \*
MERGEFORMAT">

    <w:r>
      <w:rPr>
        <w:noProof />
      </w:rPr>
      <w:t>THOMAS JEFFERSON</w:t>
    </w:r>
  </w:fldSimple>
</w:p>
```

REFERENCE:

For more information, refer to Office Open XML Part 4 - Markup Language Reference, §2.16

OFFICE 2007.4.1: **End**

OFFICE 2007.4.2: **Macros****OVERVIEW:**

MS 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. For more advanced control, users can also create or modify macros “by hand” with custom Visual Basic code. Macros either reside locally in the application or in the document or template for portability.

CONCERNS:

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

PRODUCT: WORD**LOCATION:**

MS Word 2007 documents with embedded macros must have the file extension “docm,” rather than the standard “docx.” The docm extension denotes a Word document that is capable of containing macros.

The first difference between a macro-enabled and a standard document, regardless of the presence of actual macros, is the `[Content_Types].xml` part. This part, which maps parts within the document archive to content types, will map a different content type for the `/word/document.xml` part. For macro-enabled documents, the content type for `/word/document.xml` will be `application/vnd.ms-word.document.macroEnabled.main+xml` as opposed to the default `application/vnd.openxmlformats-officedocument.wordprocessingml.document.main+xml`.

Additionally, if a macro does exist within the document archive, the line `<Default Extension="bin" ContentType="application/vnd.ms-office.vbaProject" />` will exist in the `[Content_Types].xml` part.

The next noticeable difference with a macro-enabled document is the presence of additional parts; specifically, the `/word/vbaProject.bin`, `/word/vbaData.xml`, `/word/_rels/vbaProject.bin.rels` and optional `/word/customizations.xml`. The content of the actual macro exists in the binary file `/word/vbaProject.bin`. Note that this binary file is not part of the ECMA-376 standard for Office Open XML. The `/word/customizations.xml` part will exist if a macro is mapped to a keystroke.

EXAMPLE:

`/[Content_Types].xml` (abridged):

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<Types xmlns="http://schemas.openxmlformats.org/package/2006/content-
types">
```

```

<Default Extension="bin"
ContentType="application/vnd.ms-office.vbaProject" />
<Override PartName="/word/document.xml"
ContentType="application/vnd.ms-word.document.macroEnabled.main+xml"
/>
</Types>

```

/word/vbaData.xml:

```

<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<wne:vbaSuppData xmlns:ve="http://schemas.openxmlformats.org/markup-
compatibility/2006"
xmlns:o="urn:schemas-microsoft-com:office:office"
xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relatio
nships"
xmlns:m="http://schemas.openxmlformats.org/officeDocument/2006/math"
xmlns:v="urn:schemas-microsoft-com:vml"
xmlns:wp="http://schemas.openxmlformats.org/drawingml/2006/wordprocess
ingDrawing"
xmlns:w10="urn:schemas-microsoft-com:office:word"
xmlns:w="http://schemas.openxmlformats.org/wordprocessingml/2006/main"
xmlns:rne="http://schemas.microsoft.com/office/word/2006/wordml">
  <rne:mcds>
    <rne:mcd rne:macroName="PROJECT.NEWMACROS.DEMO_MACRO"
rne:name="Project.NewMacros.demo_macro" rne:bEncrypt="00"
rne:cmg="56" />
  </rne:mcds>
</wne:vbaSuppData>

```

PRODUCT: EXCEL

LOCATION:

MS Excel 2007 documents with embedded macros must have the file extension “xlsm,” rather than the standard “xlsx.” The xlsm extension denotes an Excel document that is capable of containing macros.

The first difference between a macro-enabled and a standard document, regardless of the presence of actual macros, is the *[Content_Types].xml* part. This part, which maps parts within the document archive to content types, will map a different content type for the “/xl/workbook.xml” part. For macro-enabled documents, the content type for “/xl/workbook.xml” will be “application/vnd.ms-excel.sheet.macroEnabled.main+xml” as opposed to the default “application/vnd.openxmlformats-officedocument.spreadsheetml.sheet.main+xml”.

Additionally, if a macro exists within the document archive, the line “<Default Extension=“bin” ContentType=“application/vnd.ms-office.vbaProject” />” will exist in the *[Content_Types].xml* part.

The next noticeable difference with a macro-enabled document is the presence of additional part */xl/vbaProject.bin*. The content of the actual macro exists in this binary file.

It is also important to note that this binary file is not part of the ECMA-376 standard for OOXML.

EXAMPLE:

/[Content_Types].xml (abridged):

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<Types xmlns="http://schemas.openxmlformats.org/package/2006/content-
types">
  <Default Extension="bin"
  ContentType="application/vnd.ms-office.vbaProject" />
  <Override PartName="/xl/workbook.xml"
  ContentType="application/vnd.ms-excel.sheet.macroEnabled.main+xml"
  />
</Types>
```

PRODUCT: POWERPOINT

LOCATION:

MS PowerPoint 2007 documents with embedded macros must have the file extension “pptm,” rather than the standard “pptx.” The pptm extension denotes a PowerPoint document that is capable of containing macros.

The first difference between a macro-enabled and a standard document, regardless of the presence of actual macros, is the [Content_Types].xml part. This part, which maps parts within the document archive to content types, will map a different content type for the “/ppt/presentation.xml” part. For macro-enabled documents, the content type for “/ppt/presentation.xml” will be “application/vnd.ms-powerpoint.presentation.macroEnabled.main+xml” as opposed to the default “application/vnd.openxmlformats-officedocument.presentationml.presentation.main+xml”.

Additionally, if a macro exists within the document archive, the line “<Default Extension=“bin” ContentType=“application/vnd.ms-office.vbaProject” />” will exist in the [Content_Types].xml part.

The next noticeable difference with a macro-enabled document is the presence of additional part /ppt/vbaProject.bin. The content of the actual macro exists in this binary file. It is also important to note that this binary file is not part of the ECMA-376 standard for OOXML.

EXAMPLE:

/[Content_Types].xml (abridged):

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<Types xmlns="http://schemas.openxmlformats.org/package/2006/content-
types">
  <Default Extension="bin"
  ContentType="application/vnd.ms-office.vbaProject" />
  <Override PartName="/ppt/presentation.xml"
  ContentType="application/vnd.ms-
powerpoint.presentation.macroEnabled.main+xml" />
```

</Types>

RECOMMENDATIONS:

AR.1 Validate: Ensure consistency by validating that the macros are on a "white list" of allowable macros.

AR.2 Remove: Remove all macros from the document.

AR.3 Replace: N/A

AR.4 External Filtering Required: N/A

AR.5 Review: N/A

OFFICE 2007.4.2: **End**

OFFICE 2007.4.3: **Comments**

DESCRIPTION:

The Comments metadata field contains author or reviewer comments. MS Word and Excel support adding user comments to a document through the Review → Comments → New Comment command. In PowerPoint, users add comments as Notes in a dedicated space near the bottom of the slide.

This item applies to MS Word, Excel and PowerPoint 2007 and higher versions.

CONCERNS:

Any free-form text field provides the possibility of data disclosure threats. As such, consider comments in Office 2007 documents a data disclosure threat.

PRODUCT: WORD**LOCATION:**

Two Comment varieties are of concern. For this document, we call them "review comments" and "in-text comments".

Review Comments are notes that appear along the document margin and are generally used when editing a document. In Word, Review Comments appear in the main document along with the text. Three elements of interest exist. The elements *w:commentRangeStart* and *w:commentRangeEnd* indicate where text is highlighted in the document's main body. Presumably, this is the text or object to which the comment applies. Another element is *w:commentReference*, which exists to connect any changes in the document appearance (as a result of the comment) to the comment itself. Each of these elements contains a *w:id* attribute, which links them together and also links them to a separate comments xml file. The comment xml file contains the actual comments within

the *w:comments* element list. Each *w:comment* element has its *w:id* attribute to link it back to the original document. Beyond that, the *w:comment* element contains standard Word xml structures for containing and formatting text.

In-text Comments are notes that appear as just another type of “Quick Parts” in Word 2007. In-text Comments in Word are contained in the main document along with the text, isolated in their own *w:std* element. The *w:std* element of comments will contain a *w:stdPr* elements and (in turn) a *w:alias* element. The *w:alias* element will have an attribute *w:val* with a value of “Comments”.

RECOMMENDATION:

WR.1 Validate: Ensure consistency by verifying that all appropriate xml elements exist for the comment.

WR.2 Remove: Remove comments by deleting the appropriate comment elements and the comments xml file.

WR.3 Replace: Replace comment text by scanning for the comment elements and modifying them with the desired text.

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

WR.5 Review: N/A

EXAMPLE:

```
<!-- Review Comments in document*.xml -->
<w:p w:rsidR="002D0A8A" w:rsidRDefault="00D07A82">
  <w:r>
    <w:t xml:space="preserve">This is some text with a </w:t>
  </w:r>
  <w:commentRangeStart w:id="0" />
  <w:r>
    <w:t>comment</w:t>
  </w:r>
  <w:commentRangeEnd w:id="0" />
  <w:r>
    <w:rPr>
      <w:rStyle w:val="CommentReference" />
    </w:rPr>
    <w:commentReference w:id="0" />
  </w:r>
  <w:r>
    <w:t xml:space="preserve"> on the margin.</w:t>
  </w:r>
</w:p>

<!-- from comments*.xml -->
<w:comments>
  <w:comment w:id="0" w:author="Acme Widgets" w:date="2009-12-
09T08:27:00Z"
    w:initials="BAH">
    <w:p w:rsidR="00D07A82" w:rsidRDefault="00D07A82">
      <w:pPr>
```

```

        <w:pStyle w:val="CommentText" />
      </w:pPr>
    <w:r>
      <w:rPr>
        <w:rStyle w:val="CommentReference" />
      </w:rPr>
      <w:annotationRef />
    </w:r>
    <w:r>
      <w:t>This is the comment on the margin.</w:t>
    </w:r>
  </w:p>
</w:comment>
</w:comments>

<sdt>
  <sdtPr>
    <alias val="Comments" />
    <id val="1850407" />
    <placeholder>
      <docPart val="2F5DE0FBE3594F25B202533C80E4C554" />
    </placeholder>
    <dataBinding />
  </sdtPr>
  <sdtContent>
    <p rsidR="002325AF" rsidRDefault="002325AF">
      <r>
        <t>This is a sample comment.</t>
      </r>
    </p>
  </sdtContent>
</sdt>

```

PRODUCT: EXCEL

LOCATION:

Comments reside in Excel in dedicated comment xml documents in a *comments* element. Within the *comments* element, a *commentList* list contains individual *comment* elements. Within each *comment* element, a *text* element contains individual *r* elements, with each *r* element containing the *t* elements holding the comment strings. Comment authors also reside in individual *author* elements in an *authors* list. This list resides just below the main **comments** element.

RECOMMENDATION:

ER.1 Validate: Ensure consistency by verifying that all appropriate xml elements exist for the comment.

ER.2 Remove: Remove comments by deleting the appropriate *commentLists*. Also, remove comment authors by removing comment authors list.

ER.3 Replace: Replace comment text by scanning for comment elements and modifying them with the desired text. Replace comment authors by modifying individual author

elements with the desired text. Note that authors may also appear as text in the comments.

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

ER.5 Review: N/A

EXAMPLE:

```
<comments>
  <authors>
    <author>Acme Widgets</author>
  </authors>
  <commentList>
    <comment ref="C8" authorId="0">
      <text>
        <r>
          <t>Acme Widgets:</t>
        </r>
        <r>
          <t xml:space="preserve">
Comment at C8</t>
          </r>
        </text>
      </comment>
    </commentList>
  </comments>
```

PRODUCT: POWERPOINT

LOCATION:

Comments in PowerPoint reside in a dedicated comments directory, where individual *comment* XML files are. In the *comment* XML files, a *p:cmLst* element list holds one or more *p:cm* elements, which represent a distinct comment. A dedicated *commentAuthors* XML file also exists for comment author information. The *commentAuthors* XML file maintains an element list at *p:cmAuthorLst*. This list contains all of the comment authors in distinct *p:cmAuthor* elements.

For this document, PowerPoint comments reside in presentation notes in *notesSlide* XML documents in a *notes* list. Within the XML structure, comment text is stored in *a:t* elements. Often, one can split a single text line among multiple *a:t* elements within a single *a:p* element. Numerous *a:p* elements can exist within a single *notes* list.

RECOMMENDATION:

PR.1 Validate: Ensure consistency by verifying that all appropriate XML elements exist for the comment.

PR.2 Remove: Remove comments by deleting all comments in the *comments* XML file, comment author information in the *commentAuthors* XML file and presentation notes in the *notesSlide* xml file.

PR.3 Replace: Replace comment text by modifying comment text in the comments XML file, comment author information in the *commentAuthors* XML file and presentation notes in the *notesSlide* XM file.

PR.4 External Filtering Required: Pass all comment fields to the action configured for handling text.

PR.5 Review: N/A

EXAMPLE:

```
<!-- from comments*.xml -->
<p:cmLst>
  <p:cm authorId="0" dt="2009-09-29T13:58:21.491" idx="1">
    <p:pos x="2959" y="1589" />
    <p:text>This is a Presentation Comment.</p:text>
  </p:cm>
</p:cmLst>

<!-- from commentAuthors*.xml -->
<p:cmAuthorLst>
  <p:cmAuthor id="0" name="Acme Widgets" initials="BAH"
    lastIdx="1" clrIdx="0" />
</p:cmAuthorLst>

<!-- from notesSlide*.xml -->
<p:notes>
  <p:cSld>
    <p:spTree>
      <p:sp>
        <p:txBody>
          <a:p>
            <a:r>
              <a:t>This </a:t>
            </a:r>
            <a:r>
              <a:t>is slide 1 Notes</a:t>
            </a:r>
          </a:p>
        </p:txBody>
      </p:sp>
    </p:spTree>
  </p:cSld>
</p:notes>
```

OFFICE 2007.4.3: **End**

OFFICE 2007.4.4: **Linked Objects**

DESCRIPTION:

Linked objects are files that are attached to Office 2007 documents. The attached files can be any type. The means of attaching files to Office documents vary, but generally occur via the “Insert” options in the application.

Notes: Not all bin files in Office 2007 files should be considered linked objects (see Printer Information and Smart Tags for some examples). Therefore, wholesale removal of relationship (.rels) file content can have unintended consequences.*

Also, file type associations, stored in the XML files, may not prove accurate (the XML states that a linked object is an image, but the bin file contains a program), so one can provide additional verification using linked objects' bin files content.

CONCERNS:

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

PRODUCT: WORD

LOCATION:

In Word, linked objects appear in varied locations and settings. The element *w:object*, with an *o:OLEObject* element within, indicates a linked object. The relationships mentioned in the *r:id* attributes point to references within a relationship (*.rels) file, which reference the actual files.

EXAMPLE:

```
<!-- from *.rels -->
<Relationships

xmlns="http://schemas.openxmlformats.org/package/2006/relationships">
  <Relationship Id="rId8"
    Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/oleObject"
    Target="embeddings/oleObject2.bin" />
  <Relationship Id="rId7"
    Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/image"
    Target="media/image2.wmf" />
</Relationships>

<!-- from document*.xml -->
<w:object w:dxaOrig="5760" w:dyaOrig="269">
  <v:shape id="_x0000_i1026" type="#_x0000_t75"
    style="width:4in;height:13.5pt o:ole="">
    <v:imagedata r:id="rId7" o:title="" />
  </v:shape>
  <o:OLEObject Type="Embed" ProgID="WordPad.Document.1"
    ShapeID="_x0000_i1026" DrawAspect="Content" ObjectID="_1304920979"
```

```

        r:id="rId8" />
    </w:object>

```

PRODUCT: EXCEL

LOCATION:

In Excel, linked objects can appear in varied locations and settings. The element list *oleObjects*, with *oleObject* elements within, indicates linked objects. The relationships mentioned within the *r:id* attributes point to references within a relationship (*.rels) file, which references the actual files.

EXAMPLE:

```

<!-- from *.rels -->
<Relationships
  xmlns="http://schemas.openxmlformats.org/package/2006/relationships">
  <Relationship Id="rId3"
    Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/oleObject" Target="../embeddings/oleObject1.bin"/>
</Relationships>

<!-- from sheet*.xml -->
<oleObjects>
  <oleObject progId="Package" shapeId="1025" r:id="rId3" />
</oleObjects>

```

PRODUCT: POWERPOINT

LOCATION:

In PowerPoint, linked objects can appear in varied locations and settings. The element *p:oleObj* indicates the existence of a linked object. The relationships mentioned in the *r:id* attributes point to references within a relationship (*.rels) file, which references the actual files.

EXAMPLE:

```

<!-- from *.rels -->
<Relationships

  xmlns="http://schemas.openxmlformats.org/package/2006/relationships">
  <Relationship Id="rId3"
    Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/oleObject"
    Target="../embeddings/oleObject1.bin" />
</Relationships>

  <!-- from slide*.xml -->
  <p:oleObj spid="_x0000_sl026" name="Package" r:id="rId3"
    imgW="876240" imgH="485640" progId="Package">
    <p:embed />
  </p:oleObj>

```

RECOMMENDATIONS:

AR.1 Validate: Ensure consistency by verifying that the linked object, appropriate for the file type, appears in their *ProgID* attributes.

AR.2 Remove: Remove any linked objects where the *bin* file does not match the expected file type.

AR.3 Remove: Remove linked objects by scanning all XML for *r:id* elements to determine the attached file's location. If file is in the *embeddings* folder, delete the file and all references to it in the **.rels* file. Delete the element where the link was originally found.

AR.4 Remove: Remove linked objects by scanning all XML files for *o:OLEObject* elements. Using the *r:id* values contained there, determine the location of any attached files. Delete the files and all references to them in the **.rels* file. Also, delete the *o:OLEObject* element where the link was originally found.

AR.5 Remove: Remove all linked objects that are not explicitly referenced in the relationship (*rels*) files by scanning the media and embeddings folders and verifying them against the relationship (*rels*) files.

AR.6 Replace: N/A

AR.7 External Filtering Required: Identify the attached file or data type and pass the file or data to the action that is configured for that data type.

AR.8 Review: Present the object for review.

OFFICE 2007.4.4: **End**

OFFICE 2007.4.5: **Template Name**

DESCRIPTION:

In Word, the file name of the template to create the document resides as part of the document. This can include file path information.

This item applies to MS Word and Excel 2007 and higher versions.

CONCERNS:

File names are free-form text and can provide potential data disclosure threats. Consider template names in Office 2007 documents a data disclosure threat. File path information can include server names and user names. Template names are a possible attack consideration.

PRODUCT: WORD

LOCATION:

Template names reside in the *app* XML file in an element named *Template*. Template file paths are maintained in the *settings* XML file in an element called *w:attachedTemplate*. The *w:attachedTemplate* element is a standard relationship that references the settings XML *rels* file. The relationship points to the file path.

RECOMMENDATIONS:

WR.1 Validate: Ensure referential integrity and consistency by verifying that the *w:attachedTemplate* references a valid relationship.

WR.2 Remove: Remove template names by removing the *Template* element from the *app* XML file.

WR.2 Remove: Remove template file paths by removing the *w:attachedTemplate* element from the *settings* XML file and the relationship from the associated *settings* XML *rels* file.

WR.4 Replace: Replace template names by modifying the *Template* element in the *app* XML file with a pre-defined value.

WR.5 Replace: Replace template file paths by removing the *w:attachedTemplate* element from the *settings* XML file and the relationship from the associated *settings* XML *rels* file.

WR.6 External Filtering Required: Pass text data in the template names and paths to the action that is configured for text data.

WR.7 Review: N/A

EXAMPLE:

```
<!-- from app*.xml -->
<Properties>
  <Template>Example.dotx</Template>
</Properties>

<!-- from settings*.xml -->
<w:settings>
  <w:attachedTemplate r:id="rId1" />
</w:settings>

<!-- from settings*.xml.rels -->
<Relationships>
  <Relationship Id="rId1"
Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/attachedTemplate"
Target="file:///C:\Documents%20and%20Settings\username\My%20Documents\
2007\Example.dotx" TargetMode="External" />
</Relationships>
```

OFFICE 2007.4.5: **End**

OFFICE 2007.4.6: **Scenarios**

DESCRIPTION:

Scenarios are a MS Excel feature that allow for multiple data models. Excel supports entering multiple data models within specific spreadsheet areas (Data -> Data Tools -> What-if Analysis -

> Scenario Manager). After selecting a specific scenario, the remaining scenarios may expose data models that should not be exposed after document release to an outside party.

This applies to MS Excel 2007 and higher versions.

CONCERNS:

Any free-form text fields provide the possibility of data disclosure threats. Consider data fields (like the ones in scenarios) in Excel documents a data disclosure threat.

PRODUCT: EXCEL

LOCATION:

Scenarios in Excel reside in the *individual sheet* XML files along with cell data. When scenarios are present, the *worksheet* element will contain a *scenarios* list. Individual scenarios reside in the *scenarios* list. The current and selected scenarios (current and selected may contain different values) are indicated by the *current* and *show* attributes of the scenarios list. The numerical value, stored in the *current* and *show* attributes, reflects the indexes of the scenario within the *scenarios* list (starting at zero). When multiple scenario sets exist in the same sheet, with each set altering a different set of cells, the *current* and *show* attributes will contain the index of the last scenario for the user to select.

RECOMMENDATIONS:

ER.1 Validate: Ensure referential integrity and consistency by verifying that the individual scenario files are correctly referenced.

ER.2 Remove: Remove scenarios by deleting all scenarios lists.

ER.3 Replace: Replace all user information, contained in individual scenarios, by modifying each scenario's user attribute with a pre-determined value.

ER.4 External Filtering Required: Pass data in the scenario fields to the action that is configured for text data.

ER.5 Review: N/A

EXAMPLE:

```
<worksheet>
  <scenarios current="0" show="0">
    <scenario name="OnePlusTwo" locked="1" count="2"
      user="Acme Widgets" comment="Created by Acme Widgets on
9/18/2009">
      <inputCells r="A1" val="1" />
      <inputCells r="A2" val="2" />
    </scenario>
    <scenario name="TwoPlusThree" locked="1" count="2"
      user="Acme Widgets" comment="Created by Acme Widgets on
9/18/2009">
      <inputCells r="A1" val="2" />
      <inputCells r="A2" val="3" />
    </scenario>
    <scenario name="ThreePlusFour" locked="1" count="2"
      user="Acme Widgets" comment="Created by Acme Widgets on
9/18/2009">
      <inputCells r="A1" val="3" />
```

```

        <inputCells r="A2" val="4" />
    </scenario>
</scenarios>
</worksheet>

```

OFFICE 2007.4.6: **End**

OFFICE 2007.4.7: **Hyperlinks**

OVERVIEW:

One can create hyperlinks that contain fully qualified local paths, network share names, or email addresses. The Office hyperlink feature (Insert > Hyperlink) permits users to create links to various locations. Users also can create hyperlinks by typing or pasting a Uniform Resource Locator (URL) directly into an Office document as part of the "AutoCorrect" feature.

CONCERNS:

Hyperlinks can pose an attack risk and a data disclosure risk. The attack risk can occur through hyperlinks that take users to malicious sites or attempt to launch dangerous code on the local machine.

Data disclosure can occur several ways, including through fully qualified local paths, network paths, and e-mail addresses. These paths can provide unwanted insight into an organizations internal structure.

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

PRODUCT: WORD

LOCATION:

The hyperlink element can be a child within the *customXml* (§2.5.1.5), *fldSimple* (§2.16.21), *hyperlink* (§2.16.24), *p* (§2.3.1.22), *sdtContent* (§2.5.2.35) and *smartTag* (§2.5.1.9) elements within the *document.xml* part. The attributes and child elements describe the hyperlink types, and the link's destination, target, history, and tooltip information.

EXAMPLE:

\word\document.xml:

```

<w:hyperlink r:id="rId4" w:history="1" w:tooltip="Click here!">
  <w:r w:rsidRPr="00B36432">
    <w:rPr>
      <w:rStyle w:val="Hyperlink" />
    </w:rPr>
    <w:t>http://www.google.com</w:t>
  </w:r>
</w:hyperlink>

```

\word_rels\document.xml.rels:

```
<Relationships
xmlns="http://schemas.openxmlformats.org/package/2006/relationships">
  <Relationship Id="rId4"
Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/hyperlink
" Target="http://www.google.com" TargetMode="External" />
</Relationships>
```

REFERENCE:

For more information, refer to Office Open XML Part 4 - Markup Language Reference, §2.16.24

PRODUCT: EXCEL

LOCATION:

The hyperlink element can be a child within the *customXml* (§2.5.1.5), *fldSimple* (§2.16.21), An Excel document stores hyperlinks in each worksheet in a *hyperlinks* element list beneath the parent worksheet element. This element list holds a collection of hyperlink element children. These hyperlink elements do not contain the actual target. They contain a reference to the location in the worksheet and to a corresponding relationships file that holds the hyperlink address.

EXAMPLE:

\xl\worksheets\sheet1.xml:

```
<worksheet
xmlns="http://schemas.openxmlformats.org/spreadsheetml/2006/main"
xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships">
  <hyperlinks>
    <hyperlink ref="A1" r:id="rId1" tooltip="Google" />
    <hyperlink ref="A2" r:id="rId2" />
  </hyperlinks>
</worksheet>
```

\xl\worksheets_rels\sheet1.xml.rels:

```
<Relationships
xmlns="http://schemas.openxmlformats.org/package/2006/relationships">
  <Relationship Id="rId1"

Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/hyperlink"
Target="http://www.google.com/" TargetMode="External" />
  <Relationship Id="rId2"

Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/hyperlink"
Target="http://www.microsoft.com/" TargetMode="External" />
</Relationships>
```

REFERENCE:

For more information, refer to Office Open XML Part 4 - Markup Language Reference, §3.3.1.45

PRODUCT: POWERPOINT

LOCATION:

A PowerPoint document stores hyperlinks in each slide using the *hlinkClick* element. This element contains a reference to a corresponding relationship file that holds the hyperlink address. The *hlinkClick* element also contains the tooltip text that applies to the hyperlink.

EXAMPLE:

\ppt\slides\slide1.xml:

```
<a:p>
  <a:r>
    <a:rPr lang="en-US" dirty="0" smtClean="0">
      <a:hlinkClick r:id="rId2"
        tooltip="Click for Google" />
    </a:rPr>
    <a:t>Google, Inc.</a:t>
  </a:r>
  <a:endParaRPr lang="en-US" dirty="0" smtClean="0" />
</a:p>
```

\ppt\slides_rels\slide1.xml.rel:

```
<Relationships
xmlns="http://schemas.openxmlformats.org/package/2006/relationships">
  <Relationship Id="rId2"
Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/hyperlink"
Target="http://www.google.com/" TargetMode="External" />
</Relationships>
```

RECOMMENDATIONS:

AR.1 Validate: N/A

AR.2 Remove: Remove hyperlinks by locating the element and corresponding reference in the *rels* file. Remove links that are a fully-qualified local path, network share names, or that match a regular expression (e.g., `*yahoo.* <space> *.msn.* <space> *.google.* <space> ^https://.*`).

AR.3 Replace: Replace hyperlinks by locating the element and corresponding reference in the *rels* file. Replace links that are a fully-qualified local path, network share names, or that match a regular expression (Eg: `*yahoo.* <space> *.msn.* <space> *.google.* <space> ^https://.*`).

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

AR.5 Review: N/A

REFERENCE:

For more information, refer to Office Open XML Part 4 - Markup Language Reference, §5.1.5.3.5

OFFICE 2007.4.7:	End
------------------	------------

OFFICE 2007.4.8: **Core Properties**

OVERVIEW:

A MS Office package contains one Core File Properties part that holds common properties among Office document formats. These common properties include creator name, creation date, title, and description. One can access the Document Properties through the MS Office Button -> Prepare -> Properties.

CONCERNS:

Free-form text fields, such as core properties, provide the possibility of data disclosure.

PRODUCT: WORD, EXCEL, AND POWERPOINT

LOCATION:

A package contains one Core File Properties part at the location "*docProps/core.xml*" in the zip archive.

RECOMMENDATIONS:

AR.1 Validate: Ensure consistency by verifying that all elements in the *core* XML file validate against the schemas.

AR.2 Remove: Remove invalid items that are not found in the schema from the *core* XML file.

AR.3 Replace: Replace specific element data in the *core* XML file with predetermined harmless values.

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

AR.5 Review: N/A

EXAMPLE:

```
<cp:coreProperties
  xmlns:cp="http://schemas.openxmlformats.org/package/2006/metadata/core-properties"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:dcmitype="http://purl.org/dc/dcmitype/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <dc:title></dc:title>
  <dc:subject></dc:subject>
  <dc:creator>John Joe</dc:creator>
  <cp:keywords></cp:keywords>
  <dc:description></dc:description>
  <cp:lastModifiedBy>John Joe</cp:lastModifiedBy>
```

```

<cp:revision>15</cp:revision>
<dcterms:created xsi:type="dcterms:W3CDTF">2009-07-
28T17:35:00Z</dcterms:created>
<dcterms:modified xsi:type="dcterms:W3CDTF">2009-09-
17T13:56:00Z</dcterms:modified>
</cp:coreProperties>

```

OFFICE 2007.4.8: **End**

OFFICE 2007.4.9: **Custom Properties**

OVERVIEW:

A MS Office package may contain no more than one Custom File Properties part, which holds user created properties among Office document formats. One can access Custom Properties through MS Office Button -> Prepare -> Properties and clicking the arrow next to "Document Properties" in the Document Information Panel. Selecting "Advanced Properties" from that drop-down will open a new window for entering custom properties.

CONCERNS:

Free-form text fields, such as core properties, provide the possibility of data disclosure.

PRODUCT: WORD, EXCEL, AND POWERPOINT

LOCATION:

A package contains at most one Custom File Properties part at the location "docProps/custom.xml" in the zip archive.

RECOMMENDATIONS:

AR.1 Validate: Ensure consistency by verifying the *Custom File Properties* elements against the schema and identify any valid elements that do not appear on a configurable list of allowed elements.

AR.2 Remove: Remove items that the schema has not validated or valid elements that do not appear on the allowable list.

AR.3 Replace: N/A

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

AR.5 Review: N/A

EXAMPLE:

```

<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<Properties
xmlns="http://schemas.openxmlformats.org/officeDocument/2006/custom-
properties"

```



```

xmlns:vt="http://schemas.openxmlformats.org/officeDocument/2006/docPro
psVTTypes">
  <property fmtid="{D5CDD505-2E9C-101B-9397-08002B2CF9AE}" pid="2"
name="Alias">
    <vt:lpwstr>Test Doc</vt:lpwstr>
  </property>
  <property fmtid="{D5CDD505-2E9C-101B-9397-08002B2CF9AE}" pid="3"
name="Price">
    <vt:r8>19.99</vt:r8>
  </property>
  <property fmtid="{D5CDD505-2E9C-101B-9397-08002B2CF9AE}" pid="4"
name="Final">
    <vt:bool>true</vt:bool>
  </property>
  <property fmtid="{D5CDD505-2E9C-101B-9397-08002B2CF9AE}" pid="5"
name="Date Sold">
    <vt:filetime>2009-09-11T04:00:00Z</vt:filetime>
  </property>
</Properties>

```

OFFICE 2007.4.9: **End**

OFFICE 2007.4.10: **Extended Properties**

OVERVIEW:

A MS Office package may contain no more than one Extended File Properties part, which holds application-specific properties for each OOXML document. For example, PowerPoint documents maintain the number of hidden slides in the Extended properties. One can access Extended properties through MS Office Button -> Prepare -> Properties and clicking the arrow next to "Document Properties" in the Document Information Panel. Selecting "Advanced Properties" from the drop-down will open a new window for viewing and modifying extended properties.

CONCERNS:

Free-form text fields, such as core properties, provide the possibility of data disclosure.

PRODUCT: WORD, EXCEL AND POWERPOINT

LOCATION:

A package contains at most one Extended File Properties part at the location "*docProps/app.xml*" in the zip archive.

RECOMMENDATIONS:

AR.1 Validate: Ensure consistency by verifying the *Extended File Properties* elements against the schema and identify any valid elements that are not on a configurable list of allowed elements.

AR.2 Remove: Remove items that the schema has not validated or valid elements that do not appear on the allowable list.

AR.3 Replace: N/A

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

AR.5 Review: N/A

EXAMPLE:

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<Properties
  xmlns="http://schemas.openxmlformats.org/officeDocument/2006/extended-
  properties"
  xmlns:vt="http://schemas.openxmlformats.org/officeDocument/2006/docPro
  psVTypes">

  <TotalTime>2</TotalTime>
  <Words>0</Words>
  <Application>Microsoft Office PowerPoint</Application>
  <PresentationFormat>On-screen Show (4:3)</PresentationFormat>
  <Paragraphs>0</Paragraphs>
  <Slides>2</Slides>
  <Notes>0</Notes>
  <HiddenSlides>0</HiddenSlides>
  <MMClips>0</MMClips>
  <ScaleCrop>false</ScaleCrop>
  <HeadingPairs>
    <vt:vector size="4" baseType="variant">
      <vt:variant>
        <vt:lpstr>Theme</vt:lpstr>
      </vt:variant>
      <vt:variant>
        <vt:i4>1</vt:i4>
      </vt:variant>
      <vt:variant>
        <vt:lpstr>Slide Titles</vt:lpstr>
      </vt:variant>
      <vt:variant>
        <vt:i4>2</vt:i4>
      </vt:variant>
    </vt:vector>
  </HeadingPairs>
  <TitlesOfParts>
    <vt:vector size="3" baseType="lpstr">
      <vt:lpstr>Office Theme</vt:lpstr>
      <vt:lpstr>Slide 1</vt:lpstr>
      <vt:lpstr>Slide 2</vt:lpstr>
    </vt:vector>
  </TitlesOfParts>
  <Company>Company Name</Company>
  <LinksUpToDate>false</LinksUpToDate>
  <SharedDoc>false</SharedDoc>
  <HyperlinksChanged>false</HyperlinksChanged>
```

```
<AppVersion>12.0000</AppVersion>  
</Properties>
```

OFFICE 2007.4.10: **End**

OFFICE 2007.4.11: **Footnotes and Endnotes**

OVERVIEW:

Footnotes and endnotes are Word structures for displaying references at the end of the document or at the bottom of a page. Selecting the References tab and choosing Insert Footnote or Insert Endnote creates them.

This item applies to MS Word and Excel 2007 and higher versions.

CONCERNS:

Footnotes and endnotes are free-form text that can provide the possibility of data disclosure threats. Consider footnotes and endnotes in Office 2007 documents a data disclosure threat. In Word, footnotes and endnotes can contain the same formatting as standard text, including file inserts.

PRODUCT: WORD

LOCATION:

Footnotes are stored in a dedicated *footnotes* XML file. Endnotes reside in a dedicated *endnotes* XML file. Within each of these files, individual footnotes and endnotes, defined at *w:footnote* and *w:endnote* elements, are assigned a unique *w:id* attribute. Within the main *document* XML, these *w:id* attributes are referenced in the *w:footnoteReference* and *w:endnoteReference* elements. The reference elements form the connection between the position in the main document and the footnote or endnote content.

RECOMMENDATIONS:

WR.1 Validate: N/A

WR.2 Remove: Remove the *footnotes* XML file by scanning the *document* XML file for *w:footnoteReference* elements, and deleting those elements and the XML element that contains them.

WR.3 Remove: Remove the *endnotes* XML file by scanning the *document* XML file for *w:endnoteReference* elements, and deleting those elements and the XML element that contains them.

WR.4 Replace: Replace footnote and endnote text with alternative text.

WR.5 External Filtering Required: Pass the endnote or footnote text to the action that is configured for text data.

WR.6 Review: N/A

EXAMPLE:

```

<!-- from document*.xml -->
<w:p w:rsidR="00A40E24" w:rsidRDefault="00A40E24">
  <w:r>
    <w:rPr>
      <w:rStyle w:val="FootnoteReference" />
    </w:rPr>
    <w:footnoteReference w:id="1" />
  </w:r>
  <w:r>
    <w:rPr>
      <w:rStyle w:val="EndnoteReference" />
    </w:rPr>
    <w:endnoteReference w:id="1" />
  </w:r>
</w:p>

<!-- from footnotes*.xml -->
<w:footnotes>
  <w:footnote w:id="1">
    <w:p w:rsidR="00A40E24" w:rsidRDefault="00A40E24">
      <w:pPr>
        <w:pStyle w:val="FootnoteText" />
      </w:pPr>
      <w:r>
        <w:rPr>
          <w:rStyle w:val="FootnoteReference" />
        </w:rPr>
        <w:footnoteRef />
      </w:r>
      <w:r>
        <w:t xml:space="preserve"> This is a footnote</w:t>
      </w:r>
    </w:p>
  </w:footnote>
</w:footnotes>

<!-- from endnotes*.xml -->
<w:endnotes>
  <w:endnote w:id="1">
    <w:p w:rsidR="00A40E24" w:rsidRDefault="00A40E24">
      <w:pPr>
        <w:pStyle w:val="EndnoteText" />
      </w:pPr>
      <w:r>
        <w:rPr>
          <w:rStyle w:val="EndnoteReference" />
        </w:rPr>
        <w:endnoteRef />
      </w:r>
      <w:r>
        <w:t xml:space="preserve"> This is an endnote.</w:t>
      </w:r>
    </w:p>
  </w:endnote>

```

</w:endnotes>

OFFICE 2007.4.11: **End**OFFICE 2007.4.12: **Headers and Footers****OVERVIEW:**

Headers and footers are text and formatting users can apply to the page tops and bottoms throughout the document. In Word, users access headers and footers via Insert -> Header and Insert -> Footer commands. In Excel, users access headers and footers via a single combined Insert -> Header & Footer command.

This item applies to MS Word and Excel 2007 and higher versions.

CONCERNS:

Headers and footers are free-form text that can provide potential data disclosure threats. Consider headers and footers in Office 2007 documents a data disclosure threat. In Word, headers and footers can contain the same formatting as standard text, including file inserts.

PRODUCT: WORD**LOCATION:**

In Word, header and footer information resides in varied files. Distinct headers and footers can exist also for the first page of the document. Word also allows different headers for even numbered pages. This provides extra complexity because header and footer information may exist in these header and footer files even when headers and footers are not visible currently in the document.

Header and footer XML files contain *w:hdr* and *w:ft* elements, respectively. These flexible elements contain varied Word text structures, including relationship ids pointing to *bin* files.

When relationship ids exist in the header or footer file, dedicated header or footer *rels* files are generated to manage the relationship between the source file and the *bin* file. The standard *rels* file manages the relationship between the document XML and the *header* and *footer* XML files.

The *document* XML file contains a *w:sectPr* element, which contains a collection of *w:headerReference* and *w:footerReference* elements. The *w:headerReference* and *w:footerReference* elements contain the *relationship ids* for all header and footer files, and maintain a *w:type* attribute, which differentiates even page and first page content from the default headers and footers.

The *document* XML file's *w:sectPr* element may also contain a *w:titlePg* element when a separate first page headers and footers are used.

The *settings* XML file may contain *w:evenAndOddHeaders* and *w:evenAndOddFooters* elements to designate that dedicated even page header or footer content is to be used.

RECOMMENDATIONS:

WR.1 Validate: Ensure consistency by verifying that header and footer information in the document is actually used in the document.

WR.2 Remove: Remove the *header* and *footer* XML files and modify the appropriate relationship files, as needed

WR.3 Replace: Replace header and footer text with alternative text.

WR.4 External Filtering Required: Pass header and footer text data to the action that is configured for text data.

WR.5 External Filtering Required: Identify objects contained in the headers and footers and pass the objects to the action that is configured for the object's file or data type.

WR.6 Review: Pass images contained in headers and footers for manual human review.

EXAMPLE:

```
<!-- header*.xml files -->
<w:hdr>
  <w:p w:rsidR="00DB4131" w:rsidRDefault="00DB4131">
    <w:pPr>
      <w:pStyle w:val="Header" />
    </w:pPr>
    <w:r>
      <w:t xml:space="preserve">This is the </w:t>
    </w:r>
    <w:r w:rsidR="000B11C2">
      <w:t xml:space="preserve">odd page </w:t>
    </w:r>
    <w:r>
      <w:t>header of the document</w:t>
    </w:r>
  </w:p>
  <w:p w:rsidR="00DB4131" w:rsidRDefault="00DB4131">
    <w:pPr>
      <w:pStyle w:val="Header" />
    </w:pPr>
  </w:p>
</w:hdr>

<!-- footer*.xml files -->
<w:ftr>
  <w:p w:rsidR="00DB4131" w:rsidRDefault="00DB4131">
    <w:pPr>
      <w:pStyle w:val="Footer" />
    </w:pPr>
    <w:r>
      <w:t>This is the footer of the document</w:t>
    </w:r>
  </w:p>
</w:ftr>
```

```

        <w:pPr>
            <w:pStyle w:val="Footer" />
        </w:pPr>
    </w:p>
</w:ftr>

<!-- document*.xml files -->
<w:sectPr>
    <w:headerReference w:type="even" r:id="rId6" />
    <w:headerReference w:type="default" r:id="rId7" />
    <w:footerReference w:type="even" r:id="rId8" />
    <w:footerReference w:type="default" r:id="rId9" />
    <w:headerReference w:type="first" r:id="rId10" />
    <w:footerReference w:type="first" r:id="rId11" />
    <w:titlePg />
</w:sectPr>

<!-- header*.rels.xml files -->
<Relationships>
    <Relationship Id="rId2" Target="embeddings/oleObject1.bin" />
    <Relationship Id="rId1" Target="media/image1.wmf" />
</Relationships>

<!-- settings*.xml files -->
<w:settings>
    <w:evenAndOddHeaders/>
</w:settings>

```

PRODUCT: EXCEL

LOCATION:

Headers and footers in Excel reside in the *sheet* XML files. An element list called *headerFooter* resides in the *sheet* XML file. This element list has attributes, such as *differentOddEven* and *differentFirst*. The *differentOddEven* attribute designates if the even page headers and footers should differ from the odd pages. The *differentFirst* attribute designates if the first page headers and footers should differ from the odd pages. The *headerFooter* element list includes various header and footer elements, each explicitly named to cover the places the header or footer to use. In Excel, headers can be divided into left, center, and right sections. To maintain all of these strings in a single text element, Excel starts them with *&L*, *&C* and *&R* tags in the text.

RECOMMENDATIONS:

ER.1 Validate: Verify that header and footer information in the document is actually used in the document.

ER.2 Remove: Remove the *oddHeader*, *evenHeader*, and *firstHeader* elements from any *headerFooter* elements

ER.3 Replace: Remove text in the headers and footers with alternative text.

ER.4 External Filtering Required: Pass text data contained in the headers and footers to the action that is configured for text data.

ER.5 Review: N/A.

EXAMPLE:

```

<headerFooter differentOddEven="1" differentFirst="1">
  <oddHeader>&LLeft Odd Section Header&CCenter Odd Section
    Header&RRight Odd Section Header</oddHeader>
  <oddFooter>&LLeft Section Footer&CCenter Section
    Footer&RRight Section Footer</oddFooter>
  <evenHeader>&LLeft Even Section Header&CCenter Even Section
    Header&RRight Even Section Header</evenHeader>
  <firstHeader>&LLeft 1st Page Section Header&CCenter 1st Page
    Section Header&RRight 1st Page Section Header</firstHeader>
</headerFooter>

```

OFFICE 2007.4.12: **End**OFFICE 2007.4.13: **Hidden Data****DESCRIPTION:**

MS Office 2007 offers great flexibility in presenting data in Office files. In many cases, a document author may choose to hide data from a reader. For example, a PowerPoint presentation author may choose to omit a detailed color image from black and white printouts but keep the image in color versions of the presentation. A spreadsheet author may choose to hide cells to make the spreadsheet more readable.

This item applies to MS Word, Excel and PowerPoint, 2007 and higher versions.

Note: For the purposes of this section, "Hidden Data" does not refer to the varied ways to obfuscate or bury data in Office 2007 files. Rather, it is specific to the hidden text/cell/slide functions that are readily available in the Office 2007 GUI.

CONCERNS:

Hidden data can contain free-form text and provide the possibility of data disclosure threats. Consider hidden data in Office 2007 documents a data disclosure threat. Hidden data can also contain the same formatting as standard text, including file inserts. Consider hidden data a possible attack threat, given the variety of structures in which to hide it.

PRODUCT: WORD**LOCATION:**

In Word, the elements *w:vanish* and *w:webHidden* indicate nearby hidden text. These elements are contained in a run properties element called *w:rPr*. The difficulty comes in understanding that *w:rPr* elements enhance the element details that contain them, and these elements can vary depending on what is hidden. So, an element that contains *w:rPr* with *w:vanish* or *w:webHidden* also can contain hidden content.

Another consideration is the application behavior of the *w:vanish* and *w:webHidden* elements. They often unhide or undo the effects of a previous *w:vanish* or *w:webHidden*. This makes the elements' behavior more contextual. When implementing the

recommendations below, programs should consider that the elements *w:vanish* and *w:webHidden* actually may expose rather than hide data.

RECOMMENDATIONS:

WR.1 Validate: N/A

WR.2 Remove: Remove printer hidden information by scanning *w:Pr* elements for *w:vanish* elements.

WR.3 Remove: Remove web hidden data by scanning for *w:rPr* elements and looking for *w:webHidden* elements.

WR.4 Replace: Replace attributes that hide elements with values that do not.

WR.5 External Filtering Required: Pass text data contained in hidden elements to the action that is configured for text data.

WR.5 Review: N/A

EXAMPLE:

```
<!-- standard hidden text -->
<w:p w:rsidR="00B55AE4" w:rsidRDefault="007077A8">
  <w:r>
    <w:t xml:space="preserve">This is </w:t>
  </w:r>
  <w:r w:rsidRPr="007077A8">
    <w:rPr>
      <w:vanish />
    </w:rPr>
    <w:t xml:space="preserve">hidden </w:t>
  </w:r>
  <w:r>
    <w:t>text.</w:t>
  </w:r>
</w:p>

<!-- web hidden text -->
<w:p w:rsidR="00B55AE4" w:rsidRDefault="007077A8">
  <w:r>
    <w:t xml:space="preserve">This is </w:t>
  </w:r>
  <w:r w:rsidRPr="007077A8">
    <w:rPr>
      <w:webHidden />
    </w:rPr>
    <w:t xml:space="preserve">web hidden </w:t>
  </w:r>
  <w:r>
    <w:t>text.</w:t>
  </w:r>
</w:p>
```

PRODUCT: EXCEL

LOCATION:

One can hide data in Excel in many ways (e.g., hiding entire columns, rows, or entire sheets). One can hide cells based on the logic embedded in filter routines or in protected worksheets. The ways one can hide data in Excel seems endless. However, a relatively small set of elements exists to indicate hidden data.

One can find hidden sheets by looking in *workbook* XML files for *workbook* elements. The *workbook* element contains a *sheets* element list that contains *sheet* elements. One can find a hidden sheet where a *sheet* element has a *state* attribute with a value of *hidden*.

One can find hidden columns by looking in each *sheet* XML for the *worksheet* element. Within the *worksheet* element, find a *cols* element list that contains *col* elements. For hidden columns, find *col* elements with a *hidden* (or a *collapsed*) attribute equal to **1**.

One can find hidden rows by looking in each *sheet* xml for the *worksheet* element. Within the *worksheet* element, find a *sheetData* element list containing **row** elements. For hidden columns, find *row* elements with a *hidden* attribute (or a *collapsed* attribute) equal to **1**.

One can hide columns, rows, and individual cells in a protected workbook. Selecting “Hide” in the style options performs this action. These styles are listed in the *styles* Xml file. The *styles* Xml file contains a *styleSheet* element. Within the *styleSheet* element, a *cellXfs* element list contains *xf* elements. When users hide data using a style, the *xf* elements will contain a *protection* element with a *hidden* attribute equal to **1**. Starting at 0, so that the first *xf* element is 0 (not 1), count the index of the *xf* element. Any cell, column, or row that uses that style will reference this number. It becomes confusing here, because different elements refer to styles in different ways. Rows and cells have an *s* attribute, and columns have a *style* attribute. Locate the appropriate columns and rows using the same procedures from the previous two paragraphs. Find individual cells within the *row* elements and look at their *c* elements. Each cell has its *s* attribute there.

One can also find hidden scenarios by looking in each *sheet* XML’s *worksheet* element. Within the *worksheet* element, one can find a *scenarios* element list. The *scenarios* list contains *scenario* elements. Hidden scenarios have attributes called *hidden* with a value of **1**.

Data often hides in Excel when used for calculations that only keep the results visible. So, consider the possibility that removal or alteration of hidden data could alter visible data elsewhere in the spreadsheet.

RECOMMENDATIONS:

ER.1 Validate: N/A

ER.2 Remove: Remove hidden sheets by finding *sheet* elements in the workbook in which *state* is equal to *hidden*. Use the *relationship id* to remove the appropriate *sheet* XML file and any associated relationships that sheet may have.

ER.3 Remove: Remove hidden column data by scanning each *sheet* XML file for *col* elements with a *hidden* attribute (or a *collapsed* attribute) equal to **1**. Remove *columns hidden by styles* by scanning the *styles* XML files and looking for cell elements with *hidden* attributes.

ER.4 Remove: Remove hidden row data by scanning each *sheet* XML file for row elements with a *hidden* attribute (or a *collapsed* attribute) equal to **1**. Remove the row element and any elements it contains.

ER.5 Remove: Remove cells hidden by styles by scanning the *styles* XML files and looking for cell elements with *hidden* attributes.

ER.6 Replace: Replace attributes that hide sheets with values that do not.

ER.7 Replace: Replace attributes that hide columns with values that do not.

ER.8 Replace: Replace attributes that hide rows with values that do not.

ER.9 Replace: Replace attributes that hide cells with values that do not.

ER.10 External Filtering Required: Identify data in hidden elements and pass it to the action that is configured for that data type.

ER.11 Review: N/A

EXAMPLE:

```
<worksheet>

  <!-- hidden columns in sheet*.xml -->
  <cols>
    <col min="1" max="1" width="9.140625" customWidth="1" />
    <col min="2" max="2" width="0" hidden="1" customWidth="1" />
  </cols>

  <sheetData>

    <!-- hidden rows in sheet*.xml -->
    <row r="3" spans="1:3" hidden="1">
      <c r="A3" t="s">
        <v>3</v>
      </c>
    </row>

    <!-- Cell Hidden on Lock in sheet*.xml -->
    <row r="7" spans="1:3">
      <c r="A7" s="1" t="s">
        <v>7</v>
      </c>
    </row>

    <!-- Cell Hidden on a Collapsed Group in sheet*.xml -->
    <row r="24" spans="1:3" hidden="1" outlineLevel="1">
      <c r="A24" t="s">
        <v>14</v>
      </c>
      <c r="C24">
        <v>24</v>
      </c>
    </row>

    <!-- Cell Hidden on a Collapsed Group in sheet*.xml ver 2 -->
    <row r="26" spans="1:3" collapsed="1">
```

```

        <c r="A26" t="s">
            <v>13</v>
        </c>
        <c r="C26">
            <v>26</v>
        </c>
    </row>

    <!-- Row Height of 0 in sheet*.xml -->
    <row r="38" spans="1:4" hidden="1">
        <c r="A38" t="s">
            <v>16</v>
        </c>
    </row>

    <!-- Hidden by filter action in sheet*.xml -->
    <row r="47" spans="1:4" hidden="1">
        <c r="A47">
            <v>2</v>
        </c>
    </row>
</sheetData>

<!-- Hidden Scenarios in sheet*.xml -->
<scenarios current="3" show="3">
    <scenario name="1Plus2" locked="1" hidden="1" count="2"
        user="Acme Widgets"
        comment="Created by Acme Widgets on 10/1/2009_x000a_Modified by
Acme Widgets on 10/1/2009">
        <inputCells r="A34" val="1" />
        <inputCells r="A35" val="2" />
    </scenario>
    <scenario name="2Plus3" locked="1" count="2"
        user="Acme Widgets" comment="Created by Acme Widgets on
10/1/2009">
        <inputCells r="A34" val="2" />
        <inputCells r="A35" val="3" />
    </scenario>
    <scenario name="1Times2" locked="1" hidden="1" count="2"
        user="Acme Widgets"
        comment="Created by Acme Widgets on 10/1/2009_x000a_Modified by
Acme Widgets on 10/1/2009">
        <inputCells r="C34" val="1" />
        <inputCells r="C35" val="2" />
    </scenario>
    <scenario name="2Times3" locked="1" count="2"
        user="Acme Widgets" comment="Created by Acme Widgets on
10/1/2009">
        <inputCells r="C34" val="2" />
        <inputCells r="C35" val="3" />
    </scenario>
</scenarios>

</worksheet>

<!-- Cell Hidden on Lock in styles*.xml -->

```

```

<styleSheet>
  <cellXfs count="2">
    <xf numFmtId="0" fontId="0" fillId="0" borderId="0" xfId="0" />
    <xf numFmtId="0" fontId="0" fillId="0" borderId="0" xfId="0"
      applyProtection="1">
      <protection hidden="1" />
    </xf>
  </cellXfs>
</styleSheet>

<!-- Hidden sheets in workbook*.xml -->
<workbook>
  <sheets>
    <sheet name="Sheet4" sheetId="4" state="hidden" r:id="rId1"/>
  </sheets>
</workbook>

```

PRODUCT: POWERPOINT

LOCATION:

In PowerPoint, one may skip entire slides in presentations. Although the slide appears in PowerPoint, one would miss it in a quick run-through of the presentation. Identify these slides by scanning the various *slide* XML files and looking at their *p:sld* element. The *p:sld* element, with a *show* attribute where the value is **0**, is being skipped in a slideshow.

In PowerPoint, one may block individual content sections from printing in black and white mode. This can hide the PowerPoint content also, when in Print Preview mode or when viewing the content as it would appear in black and white. In these cases, various PowerPoint shape elements carry shape properties elements called *p:spPr*. These elements can contain a *bwMode* attribute, which controls how to handle the shape in black and white settings. When *bwMode* has a value of *hidden*, the shape element containing the *p:spPr* element is hidden in black and white settings.

RECOMMENDATIONS:

PR.1 Validate: N/A

PR.2 Remove: Remove hidden slides by finding the attribute files where the *p:sld* elements have a *show* attribute containing **0**.

PR.3 Remove: Remove hidden black and white shapes with the *bwMode* attribute set to *hidden*.

PR.3 Replace: Replace attributes that hide an element with values that do not.

PR.4 External Filtering Required: Identify the data type on hidden elements and pass the data to the action that is configured for that data type.

PR.5 Review: N/A

EXAMPLE:

```

<!-- slide*.xml example of skipping slides in a presentation -->
<p:sld showMasterSp="0" show="0" />

<!-- slide*.xml example of hiding content in Black and White settings
-->

```

```
<p:sp>
  <p:nvSpPr>
    <p:cNvPr id="3" name="Subtitle 2" />
    <p:cNvSpPr>
      <a:spLocks noGrp="1" />
    </p:cNvSpPr>
    <p:nvPr>
      <p:ph type="subTitle" idx="1" />
    </p:nvPr>
  </p:nvSpPr>
  <p:spPr bwMode="hidden" />
  <p:txBody>
    <a:bodyPr />
    <a:lstStyle />
    <a:p>
      <a:r>
        <a:rPr lang="en-US" dirty="0" />
        <a:t>Not Hidden Except in B/W</a:t>
      </a:r>
      <a:endParaRPr lang="en-US" dirty="0" />
    </a:p>
  </p:txBody>
</p:sp>
```

OFFICE 2007.4.13: **End**

OFFICE 2007.4.14: Authors**DESCRIPTION:**

Office 2007 documents use Author names in many places. Also, Office documents have many places where the author name exists as text in a document or as a database login name. However, this section focuses on author names independently of these other metadata sections.

Notes: Please consider that Comments, in particular, carry author information also. See Comments for details. Also, consider that author names reside in tracked changes. See Tracked Changes for details.

CONCERNS:

Any free-form text field provides the possibility of data disclosure threats. Consider authors in Office 2007 documents a data disclosure threat.

LOCATION:

In the *core* XML file, the main element is called *cp:coreProperties*. Within *cp:coreProperties* elements, *dc:creator* and *cp:lastModifiedBy* exist to store author information.

RECOMMENDATIONS:

AR.1 Validate: N/A

AR.2 Remove: Remove author information from the *core* XML file by finding and removing the *dc:creator* and *cp:lastModifiedBy* elements from the *cp:coreProperties* element.

AR.3 Replace: Replace author information in the *core* XML file by finding and replacing the data in the *dc:creator* and *cp:lastModifiedBy* in the *cp:coreProperties* element.

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

AR.5 Review: N/A

EXAMPLE:

```
<cp:coreProperties>
  <dc:creator>Acme Widgets</dc:creator>
  <cp:lastModifiedBy>Acme Widgets</cp:lastModifiedBy>
</cp:coreProperties>
```

OFFICE 2007.4.14: End

OFFICE 2007.4.15: Printer Information**DESCRIPTION:**

This metadata includes printer information in the document. Printer setup information often resides within a MS Word or Excel document. For network printers, this information may

include provide dangerous insight into an enterprise's internal network and less sensitive printer model names.

This item applies to MS Word, and Excel, 2007 and higher versions.

CONCERN:

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

LOCATION:

In Office 2007, printer information resides in dedicated *bin* files in the format "printerSettings*.bin". References, made to the *bin* files in the relationship files (*rels* files), connect at various points in the XML in the document.

RECOMMENDATIONS:

AR.1 Validate: N/A

AR.2 Remove: Scan all relationship (*rels*) files for relationship in which the type ends with *printerSettings*. Remove all *bin* files referenced by the relationships. Then remove the relationships from the relationship (*rels*) file.

AR.3 Replace: N/A

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

AR.5 Review: N/A

EXAMPLE:

```
<!-- An example of a relationship to a printer settings BIN file -->
<Relationships>
  <Relationship Id="rId1"
    Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/printerSettings" Target="../printerSettings/printerSettings1.bin"
  />
</Relationships>

<!-- An example of a printer settings reference in XML -->
<worksheet>
  <pageSetup orientation="portrait" r:id="rId1" />
</worksheet>
```

OFFICE 2007.4.15: **End**

OFFICE 2007.4.16: **Smart Tags**

DESCRIPTION:

This includes tags applied to text that matches a defined pattern, allowing specific actions to execute based on the smart tag category. Smart tags are an Office feature that allows association

of specific actions with text content that matches a pattern associated with each smart tag category. For example, stock ticker symbols can be recognized and tagged to make related actions available to users when encountering a ticker symbol. One locates Smart Tags by using the Office button and clicking the Applications Option button on the bottom of the window. From the Options window, choose the Add-Ins and select to Manage Smart Tags and press the "Go..." button. Excel and PowerPoint have extra checkboxes to embed the Smart Tag settings in the document, where Word simply labels each block of text found by a Smart Tag.

This item applies to MS Word, PowerPoint, and Excel, 2007 and higher versions.

CONCERNS:

Smart Tags can contain a variety of actions and be extended with third-party software. Treat Smart Tags as executable code, and therefore, consider Smart Tags data disclosure and attack threats.

PRODUCT: WORD

LOCATION:

The Smart Tag data in Word resides with each block of text found by the smart tag. From the *p* element, *smartTag* elements will appear when smart tags have occurred.

RECOMMENDATIONS:

WR.1 Validate: N/A

WR.2 Remove: Scan the XML files for *smartTag* elements. Remove the entire structure with the exception of what the *r* element covers (which will exist inside the *p* element).

WR.3 Replace: N/A

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

WR.5 Review: N/A

EXAMPLE:

```
<w:p rsidR="00BD4F85" w:rsidRDefault="00BD4F85">
  <w:smartTag w:uri="urn:schemas-microsoft-com:office:smarttags"
w:element="phone">
    <w:smartTagPr>
      <attr w:uri="urn:schemas-microsoft-com:office:office"
w:name="ls" w:val="trans" />
      <attr w:name="phonenumber" w:val="12223334444" />
    </w:smartTagPr>
    <w:r>
      <w:t>1-222-333-4444</w:t>
    </w:r>
  </w:smartTag>
</w:p>
```

PRODUCT: EXCEL

LOCATION:

In Excel, a central location indicates Smart Tag existence in the sheet. Individual cells do not carry smart tag labeling. Inside a workbook, *smartTagPr* and *smartTagTypes* elements indicate Smart Tag use.

RECOMMENDATIONS:

ER.1 Validate: N/A

ER.2 Remove: Scan the workbook XML files for *smartTagPr* and *smartTagTypes* elements and remove them.

ER.3 Replace: N/A

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

ER.5 Review: N/A

EXAMPLE:

```
<workbook>
  <smartTagPr embed="1" />
  <smartTagTypes>
    <smartTagType namespaceUri="urn:schemas-microsoft-
com:office:smartrtags"
      name="stockticker" />
  </smartTagTypes>
</workbook>
```

PRODUCT: POWERPOINT

LOCATION:

In PowerPoint, a central location indicates the existence of Smart Tags in the presentation. Individual slides do not carry Smart Tag labeling. Inside of a presentation, *smartTags* elements indicate that Smart Tag use. The *smartTags* elements contain relationships in *bin* files, which also reside in the PowerPoint file.

RECOMMENDATIONS:

PR.1 Validate: N/A

PR.2 Remove: Scan the XML files for *smartTags* elements and remove them. Follow the relationship *reference id* element to the appropriate *rels* file. Remove the relationship element from the *rels* file, and remove files referenced inside.

PR.3 Replace: N/A

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

PR.5 Review: N/A

EXAMPLE:

```
<p:presentation>
  <p:smartTags r:id="rId3" />
</p:presentation>
```

```
<Relationships>
  <Relationship Id="rId3"

  Type="http://schemas.microsoft.com/office/2006/relationships/smartTags
  "
    Target="smartTags.bin" />
</Relationships>
```

OFFICE 2007.4.16: **End**

OFFICE 2007.4.17: **Imagery**

OVERVIEW:

Use of a media folder has enhanced the ability to extract and analyze images from Office documents dramatically. Instead of embedding the images within the file, the original content of images now resides in a single folder, regardless of where the image appears within the file. A possible exception exists with embedded images that, in some cases, may reside in the embeddings folder as OLE objects. This might occur if a user chooses to use the Insert Object option instead of the Insert Picture option. Third party applications now can extract and display the images to users in original form when stored in the media folder, even if they appear cropped, as grouped objects, or have modified attributes such as brightness, contrast, or hidden within the document.

One way to reduce 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 amount of data within each image without significantly impacting the appearance of the Office document, as long as the size does not reduce to a value smaller than the visible area of the image within 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 might include PNG or JPEG. Although, increasingly, PNG is an attractive option because of the balance between file size and quality, a single file type that balances the quality and size issues for varied source image types and characteristics may not exist. Where the original image can reliably flatten, the original image type could be maintained. The goal of this document is not to require a specific target image format. Instead, 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 varied technical complexity levels. Vector images and drawings may contain additional information, such as metadata and text. This information may not be apparent to users, even if they visually inspect 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 vector images can scale without quality loss that typically results when scaling raster images. The ability to provide

enhanced quality when scaling is supported by retaining additional information about the image content. Therefore, vector images may carry additional complexities. Flattened raster image formats, with layers removed, 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 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. These drawbacks are offset by the ability to prevent data hiding behind images, or text hidden within an image that the user might not see when viewing the image.

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

CONCERNS:

Risks include all images in the document, including their alternate text, description, source, and hyperlink (if present). 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, used as watermarks or document thumbnails, or reside in the notes area of PowerPoint presentations or 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 because one can identify images with either relative coordinates or absolute coordinates. When using absolute coordinates, one can identify the object’s location easily and calculate overlapping objects relatively simply. The complexity increases when relative coordinates and (or) text wrapping is used to determine object locations. Because the object’s displayed location relies heavily on the application’s rendering engine, 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 AND POWERPOINT

LOCATION:

Shape with an image background:

Shape objects begin with a root *pict* element that contain *shapetype* and *shape* child elements. A shape object with a background image will contain a *fill* element as a child to the *shape* element. This *fill* element contains a reference attribute, *id*, which has a corresponding entry in the *relationship* file to identify the location of the external image.

Chart with an image background:

A chart that contains a background image will contain a *blipFill* element with a child *blip* element. The *child* blip will hold an attribute that stores a *relationship id* to the location of the image. Note that charts do not reside in the main document (i.e., /word/document.xml, /xl/workbook.xml). Rather, they reside in a separate file in a chart sub directory (/word/chart/chart1.xml).

SmartArt with an image background:

Similar to charts with background images, SmartArt also uses the *blipFill* element with a child *blip* element. SmartArt data resides in a series of XML files within the Diagrams parent directory. The parts that contain references to the background image are \word\diagrams\data1.xml and \word\diagrams\diagram1.xml.

RECOMMENDATIONS:

Shapes, WordArt and SmartArt with Foreground and Background Images:

AR.1.1 Validate: Ensure consistency by checking if the foreground and background image do not match. Ensure referential integrity by verifying that the foreground and background image relationships exist properly.

AR.1.2 Remove: Remove all foreground images and image overlays on text by locating the element and removing the *relationship id* of the foreground image.

AR.1.3 Remove: Remove all background images by locating the element and removing the *relationship id* of the background image.

AR.1.4 Replace: N/A

AR.1.5 External Filtering Required: Pass foreground and background images to the action that is configured for image data.

AR.1.6 Review: Pass images for manual human review.

Grouped Images:

AR.2.1 Validate: Ensure referential integrity by verifying that the image relationships properly exist.

AR.2.2 Remove: N/A

AR.2.3 Replace: Replace grouped parts into a single flattened image, and discard any non-visible data.

AR.2.4 External Filtering Required: Pass the image parts to the action that is configured for image data.

AR.2.5 Review: Pass images for manual human review.

Cropped Images:

AR.3.1 Validate: Ensure referential integrity by verifying that the image relationships properly exist.

AR.3.2 Remove: Remove images in which the cropping exceeds an allowable threshold.

AR.3.3 Replace: Replace the current image by applying the crop settings.

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

AR.3.5 Review: Pass images for manual human review.

Brightness or Contrast -Adjusted Images:

AR.4.1 Validate: Ensure referential integrity by verifying that the image relationships properly exist.

AR.4.2 Remove: Remove images in which brightness and contrast exceed an allowable threshold.

AR.4.3 Replace: Replace the current image by applying the brightness and contrast settings.

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

AR.4.5 Review: Pass images for manual human review.

Resized Images:

AR.5.1 Validate: Ensure referential integrity by verifying that the image relationships properly exist.

AR.5.2 Remove: Remove images in which resizing exceeds an allowable threshold.

AR.5.3 Replace: Replace current images by applying resizing.

AR.5.4 External Filtering Required: Pass the image to the action that is configured for image data.

AR.5.5 Review: Pass images for manual human review.

Overlapping Images:

AR.6.1 Validate: Ensure referential integrity by verifying that the image relationships properly exist.

AR.6.2 Remove: Remove non-visible portions of the overlapped image.

AR.6.3 Replace: N/A

AR.6.4 External Filtering Required: Pass the image to the action that is configured for image data.

AR.6.5 Review: Pass images for manual human review.

Document-Specific Image Metadata:

AR.7.1 Validate: Ensure consistency by validating that image metadata is legitimate.

AR.7.2 Remove: Remove alternate text, description, or source data by locating the element containing the metadata and removing the specific attributes.

AR.7.3 Replace: Replace alternate text, description, or source data by locating the element containing the metadata and replacing the specific attribute values with predefined values.

AR.7.4 External Filtering Required: Pass alternate text, description, or source data to the action that is configured for that data type.

AR.7.5 Review: Pass the text for manual human review.

EXAMPLE:

Shape with image background, WordArt with image overlay:

```
<w:pict>
  <v:shapetype id="_x0000_t202" coordsize="21600,21600"
    o:spt="202" path="m,l,21600r21600,l21600,x">
    <v:stroke jointstyle="miter" />
    <v:path gradientshapeok="t" o:connecttype="rect" />
  </v:shapetype>
  <v:shape id="_x0000_s1026" type="#_x0000_t202"
    style="position:absolute;margin-left:-.7pt;margin-
    top:4.1pt;width:211.95pt;height:125.65pt;z-index:251658240">
    <v:fill r:id="rId4" o:title="Water lilies" size="0,0"
    aspect="atLeast" origin="-32767f,-32767f"
    position="-32767f,-32767f" recolor="t" rotate="t" type="frame" />
    <v:textbox>
      <w:txbxContent>
        <w:p w:rsidR="003B5E0A"
          w:rsidRDefault="003B5E0A" />
      </w:txbxContent>
    </v:textbox>
  </v:shape>
</w:pict>
```

Chart with image background and SmartArt with image background:

```
<a:blipFill>
  <a:blip
    xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relatio
    nships" r:embed="rId1" />
```

```

<a:stretch>
  <a:fillRect />
</a:stretch>
</a:blipFill>

```

PRODUCT: WORD

LOCATION:

Insert Picture or Clip Art:

When users insert a picture or clip art image via the Insert menu, the document creates a *w:drawing* structure with many child elements and attributes that describe the location. The core elements and attributes, *blipFill* and *blip*, within this element match the previously mentioned “Chart with image background” and “SmartArt with image background.” In this format, the path to the *blipFill* element is *w:drawing/wp:inline/a:graphic/a:graphicData/pic:pic/pic:blipFill/a:blip*.

Document background:

Word documents with background images will have a *w:background* element with a *v:background* child that contains a *v:fill* child element. The *v:fill* element contains a *r:id* attribute which provides a relationship in the *rels* file to the original image.

Watermark:

Watermarks store image elements in a series of header files: */word/header1.xml*, */word/header2.xml* and */word/header3.xml*. Each header contains a *w:pict* element with the child elements *v:shapetype* and *v:shape*. The *v:shape* contains a child, *v:imagedata*, which provides a reference to the source image through the *rels* file.

Quick Parts - *IncludePicture* Field:

When a picture is included as an Include Field using the field code “IncludePicture”, the image is added to the Word document with the root element *w:pict*. This *w:pict* element holds the two common *v:shapetype* and *v:shape* elements. Although these structures resemble the previously mentioned “Shape with image background and WordArt with image overlay” structures above, they differ in how they reference the actual image. Here, the image is referred to by the *v:imagedata* element, a child of *v:shape*, through the *r:id* attribute.

WordArt with an image overlay:

WordArt with an image overlay maintains a similar structure as a shape object with an image background. The root of WordArt is a *pict* element that contains *shapetype* and *shape* elements. The *shape* element holds a child *fill* element with an *id* attribute that references the image file through the corresponding *.rels* file.

RECOMMENDATIONS:

Images Outside Page Borders:**WR.8.1 Validate:** N/A**WR.8.2 Remove:** Remove images that exceed an allowable threshold for percentage outside the page border by calculating image location relative to document boundaries.**WR.8.3 Replace:** Replace images, where portions exceed page borders with cropped images, with those portions removed.**WR.8.4 External Filtering Required:** Pass the image to the action that is configured for image data.**WR.8.5 Review:** Pass images for manual human review.**Images used in watermarks:****WR.9.1 Validate:** N/A**WR.9.2 Remove:** Remove watermark images by removing the watermark element from the document.**WR.9.3 Replace:** Replace watermark images with a predefined image by locating and modifying the watermark element.**WR.9.4 External Filtering Required:** Pass the image to the action that is configured for image data.**WR.9.5 Review:** Pass images for manual human review**EXAMPLE:**

Insert Picture or Clip Art::

```

<w:drawing>
  <wp:inline distT="0" distB="0" distL="0" distR="0">
    <wp:extent cx="5943600" cy="4457700" />
    <wp:effectExtent l="19050" t="0" r="0" b="0" />
    <wp:docPr id="1" name="Picture 0" descr="Winter.jpg" />
    <wp:cNvGraphicFramePr>
      <a:graphicFrameLocks
xmlns:a="http://schemas.openxmlformats.org/drawingml/2006/main"
noChangeAspect="1" />
    </wp:cNvGraphicFramePr>
    <a:graphic
xmlns:a="http://schemas.openxmlformats.org/drawingml/2006/main">
      <a:graphicData
uri="http://schemas.openxmlformats.org/drawingml/2006/picture">
        <pic:pic
xmlns:pic="http://schemas.openxmlformats.org/drawingml/2006/picture">
          <pic:nvPicPr>
            <pic:cNvPr id="0" name="Winter.jpg" />
            <pic:cNvPicPr />
          </pic:nvPicPr>
          <pic:blipFill>

```

```

        <a:blip r:embed="rId4" cstate="print" />
        <a:stretch>
            <a:fillRect />
        </a:stretch>
    </pic:blipFill>
    <pic:spPr>
        <a:xfrm>
            <a:off x="0" y="0" />
            <a:ext cx="5943600" cy="4457700" />
        </a:xfrm>
        <a:prstGeom prst="rect">
            <a:avLst />
        </a:prstGeom>
    </pic:spPr>
</pic:pic>
</a:graphicData>
</a:graphic>
</wp:inline>
</w:drawing>

```

Document Background:

```

<w:background w:color="FFFFFF">
    <v:background id="_x0000_s1025" o:bwmode="white"
    o:targetscreensize="800,600">
        <v:fill r:id="rId2" o:title="Blue hills" recolor="t" type="frame"
    />
    </v:background>
</w:background>

```

Watermark:

```

<w:pict>
    <v:shapetype id="_x0000_t75" coordsize="21600,21600"
    o:spt="75" o:preferrelative="t"
    path="m@4@5l@4@1l@9@1l@9@5xe" filled="f" stroked="f">
        <v:stroke jointstyle="miter" />
        <v:formulas>
            <v:f eqn="if lineDrawn pixelLineWidth 0" />
            <v:f eqn="sum @0 1 0" />
            <v:f eqn="sum 0 0 @1" />
            <v:f eqn="prod @2 1 2" />
            <v:f eqn="prod @3 21600 pixelWidth" />
            <v:f eqn="prod @3 21600 pixelHeight" />
            <v:f eqn="sum @0 0 1" />
            <v:f eqn="prod @6 1 2" />
            <v:f eqn="prod @7 21600 pixelWidth" />
            <v:f eqn="sum @8 21600 0" />
            <v:f eqn="prod @7 21600 pixelHeight" />
            <v:f eqn="sum @10 21600 0" />
        </v:formulas>
        <v:path o:extrusionok="f" gradientshapeok="t"
        o:connecttype="rect" />
        <o:lock v:ext="edit" aspectratio="t" />
    </v:shapetype>

```

```

<v:shape id="WordPictureWatermark26649481"
o:spid="_x0000_s2051" type="#_x0000_t75"
style="position:absolute;margin-left:0;margin-
top:0;width:467.9pt;height:350.9pt;z-index:-251656192;mso-position-
horizontal:center;mso-position-horizontal-relative:margin;mso-
position-vertical:center;mso-position-vertical-relative:margin"
o:allowincell="f">
  <v:imagedata r:id="rId1" o:title="Blue hills"
  gain="19661f" blacklevel="22938f" />
</v:shape>
<w:pict>

```

Quick Parts - IncludePicture Field:

```

<w:pict>
  <v:shapetype id="_x0000_t75" coordsize="21600,21600" o:spt="75"
  o:preferrelative="t" path="m@4@5l@4@1l@9@1l@9@5xe" filled="f"
  stroked="f">
    <v:stroke joinstyle="miter" />
    <v:formulas>
      <v:f eqn="if lineDrawn pixelLineWidth 0" />
      <v:f eqn="sum @0 1 0" />
      <v:f eqn="sum 0 0 @1" />
      <v:f eqn="prod @2 1 2" />
      <v:f eqn="prod @3 21600 pixelWidth" />
      <v:f eqn="prod @3 21600 pixelHeight" />
      <v:f eqn="sum @0 0 1" />
      <v:f eqn="prod @6 1 2" />
      <v:f eqn="prod @7 21600 pixelWidth" />
      <v:f eqn="sum @8 21600 0" />
      <v:f eqn="prod @7 21600 pixelHeight" />
      <v:f eqn="sum @10 21600 0" />
    </v:formulas>
    <v:path o:extrusionok="f" gradientshapeok="t" o:connecttype="rect"
  />
  <o:lock v:ext="edit" aspectratio="t" />
</v:shapetype>
<v:shape id="_x0000_i1025" type="#_x0000_t75"
style="width:207.15pt;height:82.2pt">
  <v:imagedata r:id="rId4" r:href="rId5" />
</v:shape>
</w:pict>

```

WordArt with image overlay:

See example for "Shape with image background" above.

PRODUCT: EXCEL

LOCATION:

Insert Picture or Clip Art:

When users insert a picture or clip art item in an Excel worksheet, the sheet will receive a drawing element with an *id* attribute. This element and attribute maps to the

relationships file (*.rels* file). The relationship file points to a corresponding drawing file within the drawings sub directory (*/xl/drawings/drawing1.xml*).

The drawing file contains many elements and attributes to describe the image and the common *blipFill* and *blip* elements seen in previous grammars. With the *blip* element's *embed* attribute, one can find the relationship id for the image.

Document Background:

When a background image applies to a spreadsheet in Excel, the active worksheet will have an additional picture element with corresponding *r:id*. The *r:id* maps to the relationships file to point to the location of the source image.

WordArt with image overlay or image background:

WordArt in Excel that contains a background image or image overlay will contain *blipFill* elements, each with a child *blip* element. The child *blip* will hold an attribute that stores a *relationship id* to the location of the image. Note that WordArt constructs do not reside in the current worksheet (i.e.: */xl/worksheets/sheet1.xml*) but rather in a separate file in a drawing sub directory (*/xl/drawings/drawing1.xml*).

RECOMMENDATION:

Images Outside the Default Viewable Area:

ER.10.1 Validate: N/A

ER.10.2 Remove: Remove images that exceed an allowable threshold for percentage outside the viewable area by determining the image placement in the document in the element attributes.

ER.10.3 Replace: N/A

ER.10.4 External Filtering Required: Pass the image to the action that is configured for image data.

ER.10.5 Review: Pass images for manual human review.

EXAMPLE:

Insert Picture or Clip Art:

```
/xl/worksheets/sheet1.xml:
<drawing r:id="rId1" />
```

```
/xl/drawings/drawing1.xml:
<xdr:twoCellAnchor editAs="oneCell">
  <xdr:from>
    <xdr:col>0</xdr:col>
    <xdr:colOff>0</xdr:colOff>
    <xdr:row>0</xdr:row>
    <xdr:rowOff>0</xdr:rowOff>
  </xdr:from>
```

```

<xdr:to>
  <xdr:col>12</xdr:col>
  <xdr:colOff>304800</xdr:colOff>
  <xdr:row>30</xdr:row>
  <xdr:rowOff>0</xdr:rowOff>
</xdr:to>
<xdr:pic>
  <xdr:nvPicPr>
    <xdr:cNvPr id="2" name="Picture 1" descr="Winter.jpg" />
    <xdr:cNvPicPr>
      <a:picLocks noChangeAspect="1" />
    </xdr:cNvPicPr>
  </xdr:nvPicPr>
  <xdr:blipFill>
    <a:blip
xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
      r:embed="rId1" cstate="print" />
    <a:stretch>
      <a:fillRect />
    </a:stretch>
  </xdr:blipFill>
  <xdr:spPr>
    <a:xfrm>
      <a:off x="0" y="0" />
      <a:ext cx="7620000" cy="5715000" />
    </a:xfrm>
    <a:prstGeom prst="rect">
      <a:avLst />
    </a:prstGeom>
  </xdr:spPr>
</xdr:pic>
<xdr:clientData />
</xdr:twoCellAnchor>

```

Document Background:

```
<picture r:id="rId1" />
```

WordArt with image overlay or image background:

/xl/worksheets/sheet1.xml:

```
<drawing r:id="rId1" />
```

/xl/drawings/drawing1.xml:

```

<xdr:wsDr
xmlns:xdr="http://schemas.openxmlformats.org/drawingml/2006/spreadsheetDrawing"
xmlns:a="http://schemas.openxmlformats.org/drawingml/2006/main">
  <xdr:oneCellAnchor>
    <xdr:from>
      <xdr:col>9</xdr:col>
      <xdr:colOff>599184</xdr:colOff>
      <xdr:row>14</xdr:row>
      <xdr:rowOff>71977</xdr:rowOff>

```

```

</xdr:from>
<xdr:ext cx="1392433" cy="937629" />
<xdr:sp macro="" textlink="">
  <xdr:nvSpPr>
    <xdr:cNvPr id="2" name="Rectangle 1" />
    <xdr:cNvSpPr />
  </xdr:nvSpPr>
  <xdr:spPr>
    <a:xfrm>
      <a:off x="6085584" y="2738977" />
      <a:ext cx="1392433" cy="937629" />
    </a:xfrm>
    <a:prstGeom prst="rect">
      <a:avLst />
    </a:prstGeom>
    <a:blipFill>
      <a:blip
xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relatio
nships"
        r:embed="rId1" cstate="print" />
      <a:stretch>
        <a:fillRect />
      </a:stretch>
    </a:blipFill>
  </xdr:spPr>
  <xdr:txBody>
    <a:bodyPr wrap="none" lIns="91440" tIns="45720"
rIns="91440" bIns="45720">
      <a:spAutoFit />
    </a:bodyPr>
    <a:lstStyle />
    <a:p>
      <a:pPr algn="ctr" />
      <a:r>
        <a:rPr lang="en-US" sz="5400" b="1" cap="none" spc="0">
          <a:ln w="10541" cmpd="sng">
            <a:noFill />
            <a:prstDash val="solid" />
          </a:ln>
          <a:blipFill>
            <a:blip
xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relatio
nships"
              r:embed="rId2" />
            <a:stretch>
              <a:fillRect />
            </a:stretch>
          </a:blipFill>
          <a:effectLst />
        </a:rPr>
        <a:t>Test</a:t>
      </a:r>
    </a:p>
  </xdr:txBody>
</xdr:sp>
<xdr:clientData />

```

```
</xdr:oneCellAnchor>
</xdr:wsDr>
```

PRODUCT: POWERPOINT

LOCATION:

Insert Picture or Clip Art:

Pictures and Clip Art structures within a PowerPoint presentation reside locally in each slide (*/ppt/slides/slide1.xml*). PowerPoint uses the *p:pic* element to begin the structure for the graphic. Nested within this is the common *blipFill* element with child *blip* element. This child *blip* holds an *embed* attribute, which provides a reference to the corresponding relationships (*.rels*) file. One can find the path to the source image in this *relationships* file.

Document Background:

A background image can apply to either a single slide or to all the slides within a presentation. When a background applies to a single slide, that slide (*/ppt/slides/slide1.xml*) will contain a *blipFill* element with a child *blip* element whose *embed* attribute provides a *relationship id* to the corresponding source image. However when a background image is applied to all the slides within a presentation, the */ppt/slideMasters/slideMaster1.xml* part will contain this structure instead.

WordArt with image overlay or image background:

In PowerPoint, WordArt that contains a background image or image overlay will contain *blipFill* elements, each with a child *blip* element. The child *blip* will hold an attribute that stores a *relationship id* to the location of the image. Note that WordArt grammar for PowerPoint and Excel are different than the WordArt grammar for Word. Although WordArt in Excel and PowerPoint use similar grammars, PowerPoint stores the WordArt XML structures directly in the slide.

RECOMMENDATION:

Images Outside the Page Border:

PR.11.1 Validate: N/A

PR.11.2 Remove: Remove images that exceed an allowable threshold for percentage outside the viewable area by calculating image location relative to document boundaries.

PR.11.3 Replace: Replace images, where portions exceed page borders with cropped images with those portions removed.

PR.11.4 External Filtering Required: Pass the image to the action that is configured for image data.

PR.11.5 Review: Pass images for manual human review.

EXAMPLE:

Insert Picture or Clip Art:

```

/ppt/slides/slide1.xml:
<p:sld xmlns:a="http://schemas.openxmlformats.org/drawingml/2006/main"
xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
xmlns:p="http://schemas.openxmlformats.org/presentationml/2006/main">
  <p:cSld>
    <p:spTree>
      <p:nvGrpSpPr>
        <p:cNvPr id="1" name="" />
        <p:cNvGrpSpPr />
        <p:nvPr />
      </p:nvGrpSpPr>
      <p:grpSpPr>
        <a:xfrm>
          <a:off x="0" y="0" />
          <a:ext cx="0" cy="0" />
          <a:chOff x="0" y="0" />
          <a:chExt cx="0" cy="0" />
        </a:xfrm>
      </p:grpSpPr>
      <p:pic>
        <p:nvPicPr>
          <p:cNvPr id="2" name="Picture 1" descr="Winter.jpg" />
          <p:cNvPicPr>
            <a:picLocks noChangeAspect="1" />
          </p:cNvPicPr>
          <p:nvPr />
        </p:nvPicPr>
        <p:blipFill>
          <a:blip r:embed="rId2" cstate="print" />
          <a:stretch>
            <a:fillRect />
          </a:stretch>
        </p:blipFill>
        <p:spPr>
          <a:xfrm>
            <a:off x="762000" y="571500" />
            <a:ext cx="762000" cy="571500" />
          </a:xfrm>
          <a:prstGeom prst="rect">
            <a:avLst />
          </a:prstGeom>
        </p:spPr>
      </p:pic>
    </p:spTree>
  </p:cSld>
  <p:clrMapOvr>
    <a:masterClrMapping />
  </p:clrMapOvr>
</p:sld>

```

Document Background:

/ppt/slides/slide1.xml:

```
<p:sld xmlns:a="http://schemas.openxmlformats.org/drawingml/2006/main"
xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
xmlns:p="http://schemas.openxmlformats.org/presentationml/2006/main">
```

```
  <p:cSld>
    <p:bg>
      <p:bgPr>
        <a:blipFill dpi="0" rotWithShape="1">
          <a:blip r:embed="rId2" cstate="print">
            <a:lum />
          </a:blip>
          <a:srcRect />
          <a:stretch>
            <a:fillRect />
          </a:stretch>
        </a:blipFill>
        <a:effectLst />
      </p:bgPr>
    </p:bg>
    <p:spTree>
      <p:nvGrpSpPr>
        <p:cNvPr id="1" name="" />
        <p:cNvGrpSpPr />
        <p:nvPr />
      </p:nvGrpSpPr>
      <p:grpSpPr>
        <a:xfrm>
          <a:off x="0" y="0" />
          <a:ext cx="0" cy="0" />
          <a:chOff x="0" y="0" />
          <a:chExt cx="0" cy="0" />
        </a:xfrm>
      </p:grpSpPr>
    </p:spTree>
  </p:cSld>
  <p:clrMapOvr>
    <a:masterClrMapping />
  </p:clrMapOvr>
</p:sld>
```

/ppt/slideMasters/slideMaster1.xml:

```
<p:sldMaster
xmlns:a="http://schemas.openxmlformats.org/drawingml/2006/main"
xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
xmlns:p="http://schemas.openxmlformats.org/presentationml/2006/main">
```

```
  <p:cSld>
    <p:bg>
      <p:bgPr>
        <a:blipFill dpi="0" rotWithShape="1">
          <a:blip r:embed="rId13" cstate="print">
            <a:lum />
          </a:blip>
```

```

        <a:srcRect />
        <a:stretch>
            <a:fillRect />
        </a:stretch>
    </a:blipFill>
    <a:effectLst />
</p:bgPr>
</p:bg>
<!-- Additional Content removed -->
</p:sldMaster>

```

WordArt with image overlay or image background:

/ppt/slides/slidel.xml:

```

<p:sld xmlns:a="http://schemas.openxmlformats.org/drawingml/2006/main"
xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
xmlns:p="http://schemas.openxmlformats.org/presentationml/2006/main">
    <p:cSld>
        <p:spTree>
            <p:nvGrpSpPr>
                <p:cNvPr id="1" name="" />
                <p:cNvGrpSpPr />
                <p:nvPr />
            </p:nvGrpSpPr>
            <p:grpSpPr>
                <a:xfrm>
                    <a:off x="0" y="0" />
                    <a:ext cx="0" cy="0" />
                    <a:chOff x="0" y="0" />
                    <a:chExt cx="0" cy="0" />
                </a:xfrm>
            </p:grpSpPr>
            <p:sp>
                <p:nvSpPr>
                    <p:cNvPr id="4" name="Rectangle 3" />
                    <p:cNvSpPr />
                    <p:nvPr />
                </p:nvSpPr>
                <p:spPr>
                    <a:xfrm>
                        <a:off x="3890628" y="2967335" />
                        <a:ext cx="1362745" cy="923330" />
                    </a:xfrm>
                    <a:prstGeom prst="rect">
                        <a:avLst />
                    </a:prstGeom>
                    <a:blipFill>
                        <a:blip r:embed="rId2" cstate="print" />
                        <a:stretch>
                            <a:fillRect />
                        </a:stretch>
                    </a:blipFill>
                    <a:ln>
                        <a:noFill />
                    </a:ln>
                </p:spPr>
            </p:sp>
        </p:spTree>
    </p:cSld>
</p:sld>

```

```

        </a:ln>
    </p:spPr>
    <p:txBody>
        <a:bodyPr wrap="none" lIns="91440" tIns="45720"
            rIns="91440" bIns="45720">
            <a:spAutoFit />
        </a:bodyPr>
        <a:lstStyle />
        <a:p>
            <a:pPr algn="ctr" />
            <a:r>
                <a:rPr lang="en-US" sz="5400" b="1" cap="none"
                    spc="0" dirty="0" smtClean="0">
                    <a:ln w="10541" cmpd="sng">
                        <a:noFill />
                        <a:prstDash val="solid" />
                    </a:ln>
                    <a:blipFill>
                        <a:blip r:embed="rId3" />
                        <a:stretch>
                            <a:fillRect />
                        </a:stretch>
                    </a:blipFill>
                    <a:effectLst />
                </a:rPr>
                <a:t>Text</a:t>
            </a:r>
            <a:endParaRPr lang="en-US" sz="5400" b="1" cap="none"
                spc="0" dirty="0">
                <a:ln w="10541" cmpd="sng">
                    <a:noFill />
                    <a:prstDash val="solid" />
                </a:ln>
                <a:blipFill>
                    <a:blip r:embed="rId3" />
                    <a:stretch>
                        <a:fillRect />
                    </a:stretch>
                </a:blipFill>
                <a:effectLst />
            </a:endParaRPr>
        </a:p>
    </p:txBody>
    </p:sp>
    </p:spTree>
    </p:cSld>
    <p:clrMapOvr>
        <a:masterClrMapping />
    </p:clrMapOvr>
</p:sld>

```

OFFICE 2007.4.17: **End**

OFFICE 2007.4.18: Database Connections and Queries

OVERVIEW:

MS Office supports powerful database connectivity that results in database connection and query information residing in Office documents. MS Word uses database connections to drive the Mail Merge capabilities, which include generating mailing labels from an address database. Excel provides extensive support for pulling in data from external data sources, including databases and Online Analytical Processing (OLAP) cubes. External data can display in tables, graphs or PivotTables.

CONCERNS:

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 SQL query string. The database file path or the address to a database server potentially could reveal information about an organization's internal architecture. The username 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:

Inside the document archive, the part "word\settings.xml" contains a list of database-related connection and query properties. The following list describes important properties that could contain sensitive information. These elements can exist as a child or a grandchild within the main `<w:settings>` element.

connectString (Data Source Connection String) - This element specifies the connection string to reconnect to an external data source. The string, within this element's *val* attribute, contains the connection string that the hosting application passes to an external data source access application to enable the *WordprocessingML* document to reconnect to the specified external data source. (§2.14.8)

dataSource (Data Source File Path) - This element specifies the relationship whose target is the location of the external data source to connect to a *WordprocessingML* document to perform the mail merge (for a source document) or to find the associated field data (for a merged document). (§2.14.9)

headerSource (Header Definition File Path) - This element specifies the location of a file that contains the column header information to connect to an external data source that does not have column header data specified. Specifically, this element specifies a file that corresponds with the aforementioned external data source. (§2.14.16)

linkToQuery (Query Contains Link to External Query File) - This element specifies that the current *WordprocessingML* document's query string, stored in the query element (§2.14.26)

and used to specify the data to import from the external data source, actually contains a reference to an external query file. The query file contains the actual query data to use against a specified external data source for the mail merge. This query should mimic a SQL query and be of the following form: `SELECT * FROM <query file path>`. (§2.14.18)

name (Data Source Name for Column) - This element specifies the column name in a given external data source for the column whose index is specified via the *column* element (§2.14.7). This data source name provides a column name to map a specific *MERGEFIELD* field in the document, as specified by the parent field mapping data. The *val* attribute specifies the name of this column in the data source when the connection 1 is initially established. It then permanently can link columns in the database to *MERGEFIELD* fields in the document. (§2.14.24)

odso (Office Data Source Object Settings) - This element specifies a group of additional settings for the mail merge information. It consists of an extension to the standard mail merge settings and performs two functions. First, it provides additional information about the mail merge data source. Specifically, it contains information about how to map the columns in the data source to *MERGEFIELD* fields, information about records to include and exclude when creating merged documents, and column delimiters used in text data sources. One can use this information regardless of the value of the *dataType* element (§2.14.10), when present. Second, it provides an alternate set of connection information which should be used when the *dataType* element (§2.14.10) specifies a value of native. This alternate connection string provides additional connection information for applications that support the ODSO connection string syntax. If the *dataType* element (§2.14.10) specifies that the data type of the current mail merge is not native, the second group of settings specified within this element is ignored in favor of their non-ODSO equivalents. (§2.14.25)

query (Query For Data Source Records To Merge) - This element contains the SQL string to run against the specified external data source and return the set of records from the external data, which imports into merged *WordprocessingML* documents when performing the mail merge operation. (§2.14.26)

src (ODSO Data Source File Path) - This element specifies the relationship whose target is the location of the external data source to connect to a given *WordprocessingML* document to perform the mail merge (for a source document) or find the associated field data (for a merged document) when the merge type, specified using the *dataType* element (§2.14.10), is set to native. (§2.14.30)

table (Data Source Table Name) - This element specifies a particular data set that a source or merged *WordprocessingML* document will connect to (within an external data source containing multiple data sets). In other words, when connecting a *WordprocessingML* document to an external data source that may have more than one data repository within it, (such as a database with multiple tables or a spreadsheet with multiple worksheets),

this element distinguishes the specific table or spreadsheet from which to import data from within the external data source. (§2.14.31)

udl (UDL Connection String) - This element specifies the Universal Data Link (UDL) connection string to reconnect to an external data source. The string within this element's *val* attribute contains the connection string that the host application passes to an external data source access application to enable the *WordprocessingML* document to reconnect to the specified external data source. (§2.14.34)

RECOMMENDATIONS:

WR.1 Validate: N/A

WR.2 Remove: Remove user name and passwords from the connection request.

WR.3 Remove: Remove the path to the database file or server.

WR.4 Replace: N/A

WR.5 External Filtering Required: Pass all text in the connection request to the action that is configured for text data.

WR.6 Review: N/A

EXAMPLE:

```
<w:settings xmlns:o="urn:schemas-microsoft-com:office:office"
xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
xmlns:m="http://schemas.openxmlformats.org/officeDocument/2006/math"
xmlns:v="urn:schemas-microsoft-com:vml"
xmlns:w10="urn:schemas-microsoft-com:office:word"
xmlns:w="http://schemas.openxmlformats.org/wordprocessingml/2006/main"
xmlns:sl="http://schemas.openxmlformats.org/schemaLibrary/2006/main">

  <w:zoom w:percent="110" />
  <w:proofState w:spelling="clean" w:grammar="clean" />
  <w:mailMerge>
    <w:mainDocumentType w:val="mailingLabels" />
    <w:linkToQuery />
    <w:dataType w:val="native" />
    <w:connectString w:val="Provider=MSDASQL.1;Persist Security
Info=True;Extended Properties="DSN=MS Access
Database;DBQ=C:\Documents and Settings\BAH\My
Documents\Contacts.mdb;DefaultDir=C:\Documents and Settings\BAH\My
Documents;DriverId=25;FIL=MS
Access;MaxBufferSize=2048;PageTimeout=5;";Initial
Catalog=C:\Documents and Settings\BAH\My Documents\Contacts" />
    <w:query w:val="SELECT * FROM `Contacts`" />
    <w:dataSource r:id="rId1" />
    <w:viewMergedData />
    <w:odso>
      <w:udl w:val="Provider=MSDASQL.1;Persist Security
Info=True;Extended Properties="DSN=MS Access
Database;DBQ=C:\Documents and Settings\BAH\My
Documents\Contacts.mdb;DefaultDir=C:\Documents and Settings\BAH\My
Documents;DriverId=25;FIL=MS
```

```

Access;MaxBufferSize=2048;PageTimeout=5;" ;Initial
Catalog=C:\Documents and Settings\BAH\My Documents\Contacts" />
<w:table w:val="Contacts" />
<w:src r:id="rId2" />
<w:colDelim w:val="9" />
<w:fHdr />
<w:fieldMapData>
  <w:type w:val="dbColumn" />
  <w:name w:val="ID" />
  <w:mappedName w:val="Unique Identifier" />
  <w:column w:val="0" />
  <w:lid w:val="en-US" />
</w:fieldMapData>
<w:fieldMapData>
  <w:lid w:val="en-US" />
</w:fieldMapData>
<w:fieldMapData>
  <w:type w:val="dbColumn" />
  <w:name w:val="First Name" />
  <w:mappedName w:val="First Name" />
  <w:column w:val="3" />
  <w:lid w:val="en-US" />
</w:fieldMapData>
<w:fieldMapData>
  <w:lid w:val="en-US" />
</w:fieldMapData>
<w:fieldMapData>
  <w:type w:val="dbColumn" />
  <w:name w:val="Last Name" />
  <w:mappedName w:val="Last Name" />
  <w:column w:val="2" />
  <w:lid w:val="en-US" />
</w:fieldMapData>
<w:recipientData r:id="rId3" />
</w:odso>
</w:mailMerge>
<w:defaultTabStop w:val="720" />
<w:characterSpacingControl w:val="doNotCompress" />
<w:compat />
<w:rsids>
  <w:rsidRoot w:val="001827B0" />
  <w:rsid w:val="001500E3" />
  <w:rsid w:val="001827B0" />
  <w:rsid w:val="006B190E" />
  <w:rsid w:val="00B33C52" />
  <w:rsid w:val="00E8728B" />
  <w:rsid w:val="00FC4ED8" />
</w:rsids>
<m:mathPr>
  <m:mathFont m:val="Cambria Math" />
  <m:brkBin m:val="before" />
  <m:brkBinSub m:val="--" />
  <m:smallFrac m:val="off" />
  <m:dispDef />
  <m:lMargin m:val="0" />
  <m:rMargin m:val="0" />

```

```

        <m:defJc m:val="centerGroup" />
        <m:wrapIndent m:val="1440" />
        <m:intLim m:val="subSup" />
        <m:naryLim m:val="undOvr" />
    </m:mathPr>
    <w:themeFontLang w:val="en-US" />
    <w:clrSchemeMapping w:bg1="light1" w:t1="dark1" w:bg2="light2"
w:t2="dark2" w:accent1="accent1" w:accent2="accent2"
w:accent3="accent3" w:accent4="accent4" w:accent5="accent5"
w:accent6="accent6" w:hyperlink="hyperlink"
w:followedHyperlink="followedHyperlink" />
    <w:shapeDefaults>
        <o:shapedefaults v:ext="edit" spidmax="1026" />
        <o:shapelayout v:ext="edit">
            <o:idmap v:ext="edit" data="1" />
        </o:shapelayout>
    </w:shapeDefaults>
    <w:decimalSymbol w:val="." />
    <w:listSeparator w:val="," />
</w:settings>

```

REFERENCES:

For more details, refer to Office Open XML Part 4 - Markup Language Reference, §2.14.

PRODUCT: EXCEL

LOCATION:

In the document archive, the part "*xl\connections.xml*" contains a list of external data connections to databases and other files. The following important elements could contain sensitive information. These elements can exist as a child or a grandchild within the main *<connections>* element.

connection (Connection) - This element defines how to get at an external data source and information that describes how to use the connection within the workbook. Specific constructs in a worksheet, such as OLAP formulas, QueryTables, or PivotTables, use the connection information to retrieve or refresh data based on default events or the user's explicit request. §3.13.1

dbPr (Database Properties) - This element stores all properties associated with an Open Database Connectivity (ODBC) or OLE DB external data connection. §3.13.3

olapPr (OLAP Properties) - This element contains all the properties needed for an OLAP data connection. OLE DB for OLAP is the data provider, and OLAP connections contain the *dbPr* and *olapPr* child elements. §3.13.5

textPr (Text Import Settings) - This element contains all of the text import settings. §3.13.12

webPr (Web Query Properties) - This element specifies the properties for a Web query source. A Web query will retrieve data from HTML tables, and can supply HTTP "Get"

parameters for the Web server to process (in generating the HTML by including the parameters and parameter elements). §3.13.13

The part “xl\queryTables\queryTableX.xml” (where X is an integer starting at 1) contains tables of data bound to an external query. A query table could show specific data from a text file, from a web query, or from a database query. See §3.12. The following important elements could contain sensitive information. These elements can exist as a child or a grandchild within the main *<queryTable>* element.

queryTableField (QueryTable Field) - This element holds the properties related to a specific field or column in a query table. Its parent is the *queryTableFields* element, which is found nested in the *queryTableRefresh* element. §3.12.5

deletedField (Deleted Field) - This element specifies a field that has been deleted from the query table. Its parent is the *queryTableDeletedFields* element. §3.12.1

RECOMMENDATIONS:

ER.1 Validate: N/A

ER.2 Remove: Remove user name and passwords from the connection request.

ER.3 Remove: Remove the path to the database file or server.

ER.4 Replace: N/A

ER.5 External Filtering Required: Pass all text in the connection request to the action that is configured for text data.

ER.6 Review: N/A

EXAMPLE:

\xl\connections.xml:

```
<connections
  xmlns="http://schemas.openxmlformats.org/spreadsheetml/2006/main">
  <connection id="1"
    sourceFile="C:\Documents and Settings\user\My
Documents\Contacts.mdb"
    keepAlive="1" name="Contacts" type="5" refreshedVersion="3"
    background="1" saveData="1">
    <dbPr connection="Provider=Microsoft.ACE.OLEDB.12.0;User
ID=Admin;Data Source=C:\Documents and Settings\user\My
Documents\Contacts.mdb;Mode=Share Deny Write;Extended
Properties="";Jet OLEDB:System database="";Jet
OLEDB:Registry Path="";Jet OLEDB:Engine Type=5;Jet
OLEDB:Database Locking Mode=0;Jet OLEDB:Global Partial Bulk Ops=2;Jet
OLEDB:Global Bulk Transactions=1;Jet OLEDB:New Database
Password="";Jet OLEDB:Create System Database=False;Jet
OLEDB:Encrypt Database=False;Jet OLEDB:Don't Copy Locale on
Compact=False;Jet OLEDB:Compact Without Replica Repair=False;Jet
OLEDB:SFP=False;Jet OLEDB:Support Complex Data=False"
    command="Contacts" commandType="3" />
  </connection>
</connections>
```

\xl\queryTables\queryTable1.xml:

```
<queryTable
xmlns="http://schemas.openxmlformats.org/spreadsheetml/2006/main"
name="Contacts" connectionId="1" autoFormatId="16"
applyNumberFormats="0" applyBorderFormats="0" applyFontFormats="0"
applyPatternFormats="0" applyAlignmentFormats="0"
applyWidthHeightFormats="0">
  <queryTableRefresh nextId="19">
    <queryTableFields count="18">
      <queryTableField id="1" name="ID" tableColumnId="1" />
      <queryTableField id="2" name="Company" tableColumnId="2" />
      <queryTableField id="3" name="Last Name" tableColumnId="3" />
      <queryTableField id="4" name="First Name"
tableColumnId="4" />
    </queryTableFields>
  </queryTableRefresh>
</queryTable>
```

REFERENCES:

For more details, refer to Office Open XML Part 4 - Markup Language Reference, §3.12, §3.13.

OFFICE 2007.4.18: **End**

OFFICE 2007.4.19: **Tracked Changes**

DESCRIPTION:

This item includes all tracked changes in the document. The MS Office change tracking feature tracks insertions, deletions, and formatting changes that users make to the document. 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 2007 and higher versions.

CONCERNS:

Any free-form text field provides the possibility of data disclosure threats. Consider tracked changes in Office 2007 documents a data disclosure threat.

PRODUCT: WORD

LOCATION:

Tracked changes in Word have many possible locations, based on changes made and where the changes exist in the document. However, a few ways exist to locate tracked changes by their own tags.

Inserts are contained within *w:ins* elements. Generally, one can strip the *w:ins* elements out of the document and not interfere with the final copy of the document, as long as the content within the *w:ins* element remains preserved.

Deletions reside within *w:del* elements. One can strip out the *w:del* elements along with the elements contained inside to remove historical data.

Moves are characterized by two sets of elements: moves both from and to places in the document. The originating location (from) is bordered by *w:moveFromRangeStart* and *w:moveFromRangeEnd* elements. A *w:moveFrom* element exists within these border elements. One can remove the *w:moveFrom* elements, including their internal element, but take care to remove only the *w:moveFromRangeStart* and *w:moveFromRangeEnd* elements and not the elements between them. This could result in malformed XML, as these elements serve only as floating position markers.

Similar comments pertain to the move destination. In this case, *w:moveToRangeStart* and *w:moveToRangeEnd* elements border the destination. The *w:moveTo* elements hold the change. As with *w:ins* elements, one can remove the individual *w:moveTo* elements, and the content within the elements remains to preserve the final copy of the document.

Style formatting changes are also tracked. Style elements are normally contained within *w:rPr* elements. When tracking style changes, the *w:rPr* element will contain *w:rPrChange* elements. These *w:rPrChange* elements contain the previous state of the *w:rPr* element (which can also contain *w:rPrChange* elements, and so on).

RECOMMENDATIONS:

WR.1 Validate: N/A

WR.2 Remove: Remove the historical changes and tags from the delete, move, and change tags in the document's XML files.

WR.3 Replace: Replace elements within track change elements with pre-sent values.

WR.4 External Filtering Required: Identify the type of data in track change elements and pass it to the action that is configured for that data type.

WR.5 Review: N/A

EXAMPLES:

```
<w:p w:rsidR="00EA34AE" w:rsidDel="00E754E7"
w:rsidRDefault="00EA34AE">
  <w:pPr>
    <w:rPr>
      <w:ins w:id="6" w:author="B A Hamilton" w:date="2009-09-
22T09:15:00Z" />
      <w:del w:id="7" w:author="Acme Widgets" w:date="2009-09-
22T09:26:00Z" />
    </w:rPr>
  </w:pPr>
</w:p>
```

```

<w:p w:rsidR="00EA34AE" w:rsidDel="00E754E7"
w:rsidRDefault="00EA34AE">
  <w:pPr>
    <w:rPr>
      <w:ins w:id="8" w:author="BAH" w:date="2009-09-22T09:16:00Z" />
      <w:del w:id="9" w:author="Acme Widgets" w:date="2009-09-
22T09:26:00Z" />
    </w:rPr>
  </w:pPr>

  <w:ins w:id="10" w:author="B A Hamilton" w:date="2009-09-
22T09:15:00Z">
    <w:del w:id="11" w:author="Acme Widgets" w:date="2009-09-
22T09:26:00Z">
      <w:r w:rsidDel="00E754E7">
        <w:delText>This is a 3</w:delText>
      </w:r>
      <w:r w:rsidR="003E70A3" w:rsidRPr="003E70A3">
        <w:rPr>
          <w:vertAlign w:val="superscript" />
          <w:rPrChange w:id="12" w:author="B A Hamilton"
            w:date="2009-09-22T09:15:00Z">
            <w:rPr />
          </w:rPrChange>
        </w:rPr>
        <w:delText>rd</w:delText>
      </w:r>
      <w:r w:rsidDel="00E754E7">
        <w:delText xml:space="preserve" />
      </w:r>
    </w:del>
  </w:ins>

  <w:ins w:id="13" w:author="Acme Widgets" w:date="2009-09-
22T09:44:00Z">
    <w:r w:rsidR="00062DA5">
      <w:t>5</w:t>
    </w:r>
    <w:r w:rsidR="00062DA5" w:rsidRPr="00062DA5">
      <w:rPr>
        <w:vertAlign w:val="superscript" />
        <w:rPrChange w:id="14" w:author="Acme Widgets"
          w:date="2009-09-22T09:44:00Z">
          <w:rPr />
        </w:rPrChange>
      </w:rPr>
      <w:t>th</w:t>
    </w:r>
    <w:r w:rsidR="00062DA5">
      <w:t xml:space="preserve" />
    </w:r>
  </w:ins>

  <w:ins w:id="15" w:author="B A Hamilton" w:date="2009-09-
22T09:15:00Z">

```

```

        <w:del w:id="16" w:author="Acme Widgets" w:date="2009-09-
22T09:26:00Z">
            <w:r w:rsidDel="00E754E7">
                <w:delText>layer of text.</w:delText>
            </w:r>
        </w:del>
    </w:ins>
</w:p>

<w:p w:rsidR="00454771" w:rsidRDefault="00454771" w:rsidP="00454771">
    <w:moveToRangeStart w:id="1" w:author="Acme Widgets"
        w:date="2009-09-22T11:08:00Z" w:name="move241381063" />
    <w:moveTo w:id="2" w:author="Acme Widgets" w:date="2009-09-
22T11:08:00Z">
        <w:r>
            <w:t>This is a 4</w:t>
        </w:r>
        <w:r w:rsidRPr="00D03E15">
            <w:rPr>
                <w:vertAlign w:val="superscript" />
            </w:rPr>
            <w:t>th</w:t>
        </w:r>
        <w:r>
            <w:t xml:space="preserve"> layer of text.</w:t>
        </w:r>
    </w:moveTo>
</w:p>
<w:moveToRangeEnd w:id="1" />

<w:p w:rsidR="00305B49" w:rsidDel="00454771"
w:rsidRDefault="00305B49">
    <w:pPr>
        <w:rPr>
            <w:ins w:id="20" w:author="BAH" w:date="2009-09-22T09:16:00Z" />
        </w:rPr>
    </w:pPr>
    <w:moveFromRangeStart w:id="21"
        w:author="Acme Widgets" w:date="2009-09-22T11:08:00Z"
        w:name="move241381063" />
    <w:moveFrom w:id="22" w:author="Acme Widgets"
        w:date="2009-09-22T11:08:00Z">
        <w:ins w:id="23" w:author="BAH" w:date="2009-09-22T09:16:00Z">
            <w:r w:rsidDel="00454771">
                <w:t>This is a 4</w:t>
            </w:r>
            <w:r w:rsidR="00D03E15" w:rsidRPr="00D03E15"
w:rsidDel="00454771">
                <w:rPr>
                    <w:vertAlign w:val="superscript" />
                    <w:rPrChange w:id="24" w:author="BAH"
                        w:date="2009-09-22T09:16:00Z">
                        <w:rPr />
                    </w:rPrChange>
                </w:rPr>
            <w:t>th</w:t>

```

```

        </w:r>
        <w:r w:rsidDel="00454771">
            <w:t xml:space="preserve"> layer of text.</w:t>
        </w:r>
    </w:ins>
</w:moveFrom>
</w:p>
<w:moveFromRangeEnd w:id="21" />

<!-- Tracked Style change in document*.xml -->
<w:r w:rsidRPr="00363D24">
    <w:rPr>
        <w:b />
        <w:rPrChange w:id="1" w:author="Booz Allen Hamilton"
            w:date="2009-12-09T08:31:00Z">
            <w:rPr />
        </w:rPrChange>
    </w:rPr>
    <w:t>Bold</w:t>
</w:r>

```

PRODUCT: EXCEL

LOCATION:

Tracked changes in Excel reside in a dedicated folder within the Excel document. The *revisions* folder contains all of the change and author information in the spreadsheet in a set of XML files. The files include usernames, revisionHeaders, a relationship file and one or more log files containing the actual changes.

RECOMMENDATIONS:

ER.1 Validate: N/A

ER.2 Remove: Remove the historical changes and tags from the delete, move, and change tags in the document's XML files.

ER.3 Replace: Replace elements within track change elements with pre-sent values.

ER.4 External Filtering Required: Identify the type of data in track change elements and pass it to the action that is configured for that data type.

ER.5 Review: N/A

EXAMPLE:

```

<headers>
    guid="{4AB1F0B1-1A17-4490-AEA4-863C1A7C5096}" diskRevisions="1"
    revisionId="3" version="4">
    <header guid="{4AB1F0B1-1A17-4490-AEA4-863C1A7C5096}"
    dateTime="2009-09-22T09:54:14"
        maxSheetId="4" userName="Acme Widgets" r:id="rId4" minRid="3"/>
</headers>

<revisions>
    <rcc rId="3" sId="1">

```

```

<oc r="A4" t="inlineStr">
  <is>
    <t>This is change 3 at A4</t>
  </is>
</oc>
<nc r="A4" t="inlineStr">
  <is>
    <t>This is change 4 at A4</t>
  </is>
</nc>
</rcc>
<rcv guid="{1C802F79-ED82-44BD-A6C6-8AC1859DEC0A}" action="delete"
/>
<rcv guid="{1C802F79-ED82-44BD-A6C6-8AC1859DEC0A}" action="add" />
</revisions>

<Relationships>
  <Relationship Id="rId4"/>
</Relationships>

```

OFFICE 2007.4.19: **End**

OFFICE 2007.4.20: **Document Security**

DESCRIPTION:

Document protections refer to any security features to apply to documents to protect content from viewing or modification. Protection forms include simple write protection, read protection where the document is encrypted with Standard, Agile or Extensible Encryption, or digital signatures to ensure that content remains unaltered.

CONCERNS:

Encryption:

Office 2007 documents (ECMA-376) that are encrypted using Standard Encryption are vulnerable to certain types of attacks. However, the recommendation is to use Agile Encryption, with AES and SHA-2 algorithms, to avoid security concerns. Microsoft addresses security issues with the ECMA-376 encryption algorithms in the Security Concerns section of their MS-OFFCRYPTO document:

“ECMA-376 Document encryption using Standard Encryption does not support cipher block chaining, and does not have a provision for detecting corruption, though a block cipher (specifically AES) is not known to be subject to bit flipping attacks. ECMA-376 documents using Agile Encryption are required to use cipher block chaining and corruption detection, and are not subject to the issues noted for Standard Encryption.”

[MS-OFFCRYPTO].pdf - 4.3.1 ECMA-376 Document Encryption

Digital Signatures:

Besides the security issues associated with password protecting documents, digital signatures do not provide strong protection to prevent users from sneaking data inside the document. Certain parts of the archive are not subject to signing, such as *core.xml* and *app.xml*. Users can modify these parts without invalidating the signature.

Write Reservation Password:

Concerning write-protection, where a password is required to edit an Office document, passwords are hashed and stored in the *\word\settings.xml* file. Although write-protection exists to prevent accidental editing, removing the password manually will grant the user write-access to the file.

PRODUCT: WORD, EXCEL AND POWERPOINT

LOCATION:

Encryption:

Encryption within OOXML documents is not part of the ECMA-376 standard; however, vendor-specific encryption still remains available. MS Office 2007 provides document encryption in one of three methods:

Standard Encryption – this approach uses a binary *EncryptionInfo* structure, AES as an encryption algorithm, and SHA-1 as a hashing algorithm.

Agile Encryption – this approach uses an XML *EncryptionInfo* structure. The structure specifies the encryption and hashing algorithms and can use any encryption that the host computer supports.

Extensible Encryption – this approach uses an extensible mechanism to use arbitrary cryptographic modules.

[MS-OFFCRYPTO].pdf, Section 1.3.3 Encryption

Encrypted documents contain the entire document as a single stream in a OLE compound file. The OLE compound file structure can be found in [MS-OFFCRYPTO].pdf, Section 1.3.1.

Digital Signatures:

When users apply digital signatures to a document, either through Insert tab > Signature Line > Microsoft Office Signature Line or through Microsoft Office Button > Prepare > Add a Digital Signature, a new path is created in the document archive at the location *_xmlsignatures*. This new path contains *sig1.xml*, *origins.sig* and its own *_rels* directory, which holds a corresponding relationship file for *origins.sig*.

RECOMMENDATIONS:**Encryption:****AR.1 Validate:**

AR.2 Remove: Remove encrypted components.

AR.3 Replace: Replace encrypted components with clear-text versions of the components, if decryption is possible.

AR.4 External Filtering Required: Pass clear-text versions of the components to the action that is configured for the component's data type.

AR.5 Review: N/A

AR.6 Reject: Fail documents that contain encrypted components.

Digital Signatures:

AR.1 Validate: Verify the digital signature, if possible.

AR.2 Remove: Remove digital signatures from document by deleting the *_xmldsignatures* path in the file and adjusting the appropriate relationship files, as needed.

AR.3 Replace: N/A

AR.4 External Filtering Required: N/A

AR.5 Review: N/A

AR.6 Reject: Fail documents that contain digital signatures.

EXAMPLE:**Encryption:**

See [MS-OFFCRYPTO].pdf, Section 1.3.1, for an illustration of the Data Spaces structure used when storing content in OLE compound files.

Digital Signatures:

/_xmldsignatures/sig1.xml: (abridged)

```
<?xml version="1.0" encoding="utf-8"?>
<Signature Id="idPackageSignature"
xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>
    <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-
xml-c14n-20010315" />
    <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-
sha1" />
    <Reference URI="#idPackageObject"
Type="http://www.w3.org/2000/09/xmldsig#Object">
      <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"
/>
      <DigestValue>kzaEXMJpjU0enoSoYVzDhma/gKM=</DigestValue>
</Reference>
```

```
<Reference URI="#idOfficeObject"
Type="http://www.w3.org/2000/09/xmldsig#Object">
  <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"
/>
  <DigestValue>272jhGyWADmLY2eFGfzkhi j/VXk=</DigestValue>
</Reference>
</SignedInfo>
<SignatureValue>LRnimfs6Q...H0Nl4w==</SignatureValue>
<KeyInfo>
  <KeyValue>
    <RSAKeyValue>
      <Modulus>xzzMDft...EJN6Q==</Modulus>
      <Exponent>AQAB</Exponent>
    </RSAKeyValue>
  </KeyValue>
  <X509Data>
    <X509Certificate>MIIEFD...UaIT0/g==</X509Certificate>
  </X509Data>
</KeyInfo>
<Object Id="idPackageObject"
xmlns:mdssi="http://schemas.openxmlformats.org/package/2006/digital-
signature">
  <Manifest>
    <Reference
URI="/word/document.xml?ContentType=application/vnd.openxmlformats-
officedocument.wordprocessingml.document.main+xml">
      <DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
      <DigestValue>WjMkcf91JEa4FZEsbtYzCWBKNw=</DigestValue>
    </Reference>
    <Reference
URI="/word/fontTable.xml?ContentType=application/vnd.openxmlformats-
officedocument.wordprocessingml.fontTable+xml">
    </Manifest>
  <SignatureProperties>
    <SignatureProperty Id="idSignatureTime"
Target="#idPackageSignature">
      <mdssi:SignatureTime>
        <mdssi:Format>YYYY-MM-DDThh:mm:ssTZD</mdssi:Format>
        <mdssi:Value>2009-10-16T13:53:30Z</mdssi:Value>
      </mdssi:SignatureTime>
    </SignatureProperty>
  </SignatureProperties>
</Object>
<Object Id="idOfficeObject">
  <SignatureProperties>
    <SignatureProperty Id="idOfficeV1Details"
Target="#idPackageSignature">
      <SignatureInfoV1
xmlns="http://schemas.microsoft.com/office/2006/digsig">
        <SetupID></SetupID>
        <SignatureText></SignatureText>
        <SignatureImage></SignatureImage>
        <SignatureComments>Approved by Joe
User.</SignatureComments>
        <WindowsVersion>5.1</WindowsVersion>
```

```

<OfficeVersion>12.0</OfficeVersion>
<ApplicationVersion>12.0</ApplicationVersion>
<Monitors>1</Monitors>
<HorizontalResolution>1400</HorizontalResolution>
<VerticalResolution>747</VerticalResolution>
<ColorDepth>32</ColorDepth>
<SignatureProviderId>
{00000000-0000-0000-0000-000000000000}</SignatureProviderId>
<SignatureProviderUrl></SignatureProviderUrl>
<SignatureProviderDetails>9</SignatureProviderDetails>
<ManifestHashAlgorithm>

http://www.w3.org/2000/09/xmldsig#sha1</ManifestHashAlgorithm>
<SignatureType>1</SignatureType>
</SignatureInfoV1>
</SignatureProperty>
</SignatureProperties>
</Object>
</Signature>

```

REFERENCES:

Open Office XML Part 2 - Open Packaging Conventions; §12.2 Digital Signature Parts

[MS-OFFCRYPTO].pdf - 1.3.1 Data Spaces; 1.3.3 Encryption

PRODUCT: WORD**LOCATION:**

Write Reservation Password:

Documents can contain edit protection in one of two ways. First, the document can contain the *w:writeProtection* element, which resides in the \word\settings.xml file in the document archive. If this exists, a password is hashed and is required to make any changes to the document.

The second form of document protection exists with the *w:documentProtection* element within the \word\settings.xml file in the document archive. This method gives users more control over modifiable document parts. For example, one can restrict editing to certain domain users and to certain actions, such as tracked changes, comments, and filling in forms. When this occurs, additional elements, *w:permStart* and *w:permEnd*, will exist in \word\document.xml to denote access permissions and actions available.

EXAMPLE:

Write Reservation Password:

For *w:writeProtection*:

\word\settings.xml

```

<w:writeProtection w:cryptProviderType="rsaFull"
w:cryptAlgorithmClass="hash" w:cryptAlgorithmType="typeAny"
w:cryptAlgorithmSid="4" w:cryptSpinCount="100000"

```

```
w:hash="zzo33o2ClSNnudKIMsxWYDoEjcA="
w:salt="j4HLFD7Jvg/P2xiYepv+vg==" />
```

For w:documentProtection:

\word\settings.xml

```
<w:documentProtection w:edit="readOnly" w:enforcement="0" />
```

\word\document.xml

```
<w:p w:rsidR="00FC4ED8" w:rsidRDefault="007952A2">
  <w:r>
    <w:t>This is a protected document.</w:t>
  </w:r>
  <w:permStart w:id="0" w:edGrp="everyone" />
  <w:permEnd w:id="0" />
</w:p>
```

REFERENCE:

Open Office XML Part 4 - Markup Language Reference; §2.15.1.94 writeProtection;
§2.15.1.28 documentProtection

PRODUCT: EXCEL

LOCATION:

Write Reservation Password:

One can perform write-protection at the workbook and worksheet levels. At the workbook level, the \xl\workbook.xml part contains a *fileSharing* element with an optional *reservationPassword* attribute. At the worksheet level, the part \xl\worksheets\sheet1.xml contains the element *sheetProtection* with attributes that describe how the protection is applied to the sheet.

Note: Using Excel 2007's other protection features like "Protect and Share Workbook" and "Protect Workbook - Protect Structure and Windows" from the Review tab will cause the document to reside in an OLE compound container format. For more details on the OLE compound format see [MS-OFFCRYPTO].pdf, Section 1.3.1.

EXAMPLE:

Write Reservation Password:

\xl\workbook.xml:

```
<fileSharing userName="joeuser" reservationPassword="CF3D" />
```

\xl\worksheet.xml:

```
<sheetProtection password="CF3D" sheet="1" objects="1" scenarios="1"
/>
```

REFERENCE:

Open Office XML Part 4 - Markup Language Reference; §3.2.12 fileSharing

PRODUCT: POWERPOINT**LOCATION:**

Write Reservation Password:

PowerPoint documents that contain password protection have the *modifyVerifier* element in the \ppt\presentation.xml part.

This element's attributes specify the write-protection settings that are applied to the document, including the hash algorithm used and the password's hash value.

RECOMMENDATIONS:**EXAMPLE:**

Write Reservation Password:

```
<p:presentation
xmlns:a="http://schemas.openxmlformats.org/drawingml/2006/main"
xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
xmlns:p="http://schemas.openxmlformats.org/presentationml/2006/main"
saveSubsetFonts="1">
  <p:modifyVerifier cryptProviderType="rsaFull"
    cryptAlgorithmClass="hash" cryptAlgorithmType="typeAny"
    cryptAlgorithmSid="4" spinCount="100000"
    saltData="1WHe5oNZlfYdScYRykNuww=="
    hashData="l6vTwu9tOcrDwAcKjs/+3TWjPPg=" />
</p:presentation>
```

RECOMMENDATIONS:

Write Reservation Password:

AR.1 Validate: N/A

AR.2 Remove: Remove write reserved password fields from document by deleting the appropriate elements from the client-specific XML paths.

AR.3 Replace: N/A

AR.4 External Filtering Required: N/A

AR.5 Review: N/A

AR.6 Reject: Fail documents that write reserved passwords.

REFERENCE:

Open Office XML Part 4 - Markup Language Reference; §4.3.1.17 modifyVerifier
(Modification Verifier)

OFFICE 2007.4.20: **End**

OFFICE 2007.4.21: Custom Markup**OVERVIEW:**

The OOXML standard describes support for Custom XML and Structured Document Tags, features which allow user-defined XML markup to exist within a document. When embedding Custom XML in a document, user-supplied schemas need to be provided to validate the custom markup. These custom schemas supplement the existing *WordprocessingML*, *SpreadsheetML* and *PresentationML* schemas that are core to the OOXML standard. According to Microsoft:

“[Office] Open XML allows for *custom* XML markup within the body of a document which is a handy way to allow users to tag their content for interoperability with other types of software such as a custom line of business system.”

<http://www.microsoft.com/interop/letters/ChrisCapOpenLetter.mspx>

In fact, Microsoft uses Custom XML to store version and user information in files that reside on servers running SharePoint, a collaboration and web-publishing platform.

Custom XML support also allows enhanced backwards compatibility support. This becomes apparent if the OOXML standard changes or grows and new elements and attributes are introduced. Future documents, valid against the new standard, still could achieve some compatibility level with legacy consuming applications, like the previous versions of Word, Excel or PowerPoint.

In addition to Custom XML, Structured Document Tags is a feature that allows interaction between users and the embedded customer-defined data. These “content controls” allow users to define the data’s behavior and appearance. For example, a date picker could occur inline within a document that provides a pop-up calendar for selecting dates.

Finally custom markup can exist validly in documents through schemas that support arbitrary markup. For example, the *xsd:any* tag, found within certain elements throughout the schemas, allows for child sub-elements. File inspection and sanitization programs must remain aware of the elements and attributes that allow custom markup and should filter these elements and attributes accordingly.

CONCERNS:

Custom XML poses a wide array of data disclosure concerns. First, when Custom XML markup exists within a document, the consumer application (Word, Excel and PowerPoint) will not validate the XML strictly against schemas. Therefore, the consuming application simply will bypass non-standard elements. The validation will only occur on recognized elements and attributes defined within the schemas. This means that a well-formed document with Custom XML will open without issue in a consuming application and the application will only render recognized markup. The data disclosure risk is high with Custom XML because a typical user will not notice the Custom XML markup embedded in the document when simply viewing the document in an application. When saving a document with custom markup in an application that lacks provided Custom XML schemas, the application will remove the custom markup and only write the supported elements and attributes to the file.

The data disclosure issue becomes more apparent in SharePoint-stored files. SharePoint adds several XML files to the document (in the zip archive) that contain information tags such as *lastModifiedBy*, which will expose user information. Sanitizing other metadata elements for user information may not remove the SharePoint custom markup.

PRODUCT: WORD, EXCEL, AND POWERPOINT

LOCATION:

Custom XML markup can occur in any well-formed XML part throughout a document archive.

SharePoint adds at least six files to the *customXML* directory at the top level of the zip archive: *itemProps1.xml*, *itemProps2.xml*, *itemProps3.xml*, *item1.xml*, *item2.xml*, *item3.xml*. Only the *itemProps* files are listed in the *[Content Types].xml*.

RECOMMENDATIONS:

AR.1 Validate: Validate the custom markup if an XML schema is available.

AR.2 Remove: Remove custom XML by deleting the elements and ensuring that the remaining XML is still valid.

AR.3. Remove: Remove SharePoint files from the *customXML* directory at the top level of document archive.

AR.4 Replace: Replace data in SharePoint XML that exposes user information with blank or empty strings.

AR.4 External Filtering Required: Identify data in the *custom XML* and pass to the action that is configured for that data type.

AR.5 Review: N/A

EXAMPLE:

/word/document.xml:

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<w:document xmlns:ve="http://schemas.openxmlformats.org/markup-
compatibility/2006"
xmlns:o="urn:schemas-microsoft-com:office:office"
xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relatio
nships"
xmlns:m="http://schemas.openxmlformats.org/officeDocument/2006/math"
xmlns:v="urn:schemas-microsoft-com:vml"
xmlns:wp="http://schemas.openxmlformats.org/drawingml/2006/wordprocess
ingDrawing"
xmlns:w10="urn:schemas-microsoft-com:office:word"
xmlns:w="http://schemas.openxmlformats.org/wordprocessingml/2006/main"
xmlns:wne="http://schemas.microsoft.com/office/word/2006/wordml">
  <w:body>
    <w:p w:rsidR="00FC4ED8" w:rsidRDefault="00742653">
      <w:r>
        <w:t>Hello World!</w:t>
```



```

        <myCustomMarkup date="01/01/2010">This Custom XML is not
        visible to the user.</myCustomMarkup>
    </w:r>
</w:p>
<w:sectPr w:rsidR="00FC4ED8" w:rsidSect="00FC4ED8">
    <w:pgSz w:w="12240" w:h="15840" />
    <w:pgMar w:top="1440" w:right="1440" w:bottom="1440"
    w:left="1440" w:header="720" w:footer="720" w:gutter="0" />
    <w:cols w:space="720" />
    <w:docGrid w:linePitch="360" />
</w:sectPr>
</w:body>
</w:document>

```

OFFICE 2007.4.21: **End**

OFFICE 2007.4.22: **Color & Size Obfuscation**

OVERVIEW:

Some characters are visually obscured because of the font color matching the background color. The font color of some document text closely matches the background color of the text, which results in text that is not visible in the authoring application.

Some character sizes fall outside a certain normal ranges (i.e., some character sizes fall below the value defined by the Size Obfuscated Text Minimum or above the value defined by Size Obfuscated Text Maximum).

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

CONCERNS:

Any free-form text fields provide the possibility of data disclosure threats. One should consider color and size obfuscation instances 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 text obfuscation.

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 (where opacity cause text obfuscation) such that the text becomes visible.

AR.8 Replace: Readjust the size of text, where the 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 the data type.

AR.10 Review: Adjust text with color or opacity obfuscation and submit it 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 and submit them for human review. Accounting for scaling and cropping.

OFFICE 2007.4.22: **end**

5. XML SCHEMA CONSTRUCTS

Office 2007 documents use XML for the bulk of their file structure. A benefit of XML is that the format provides formalized ways to validate that a document's XML structure adheres to standards specified in a schema. An XML schema refers to the rules defining the expected element names, values, and attributes in an XML file. In this manner, an Office 2007 file is "valid" if its XML can be validated against Microsoft's provided schema for Office 2007 files. For a basic understanding of XSDs and the part they play in validating XML, please visit Wikipedia at the following URL:

<http://en.wikipedia.org/wiki/Xsd>

One can find the Office 2007 schema on the Web at the following URL:

http://www.ecma-international.org/news/TC45_current_work/TC45_available_docs.htm

The Microsoft-provided schema appears in this document often. However, developers have altered the schema to change their focus. This was an attempt at showing only those XML elements that are "interesting" to the subject matter.

For example, the Microsoft schema goes into great detail about the many ways to represent formatted text in Word. If the current section of the document is specific to Comments in Word, the provided schema are edited to focus on the question "What is the structure of Comments in Word 2007?" without getting lost in defining the many other XML elements that can appear inside a *w:p* element.

Many of the Office 2007 constructs use similar data structures, especially for low level data such as strings and integers. This section defines some of those common structures.

Shared Schema

```
<xs:complexType name="SimpleLiteral">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      This is the default type for all of the DC elements.
      It permits text content only with optional
      xml:lang attribute.
      Text is allowed because mixed="true", but sub-elements
      are disallowed because minOccurs="0" and maxOccurs="0"
      are on the xs:any tag.

      This complexType allows for restriction or extension permitting
      child elements.
    </xs:documentation>
  </xs:annotation>
</xs:complexType>
```

```

</xs:annotation>

<xs:complexContent mixed="true">
  <xs:restriction base="xs:anyType">
    <xs:sequence>
      <xs:any processContents="lax" minOccurs="0" maxOccurs="0" />
    </xs:sequence>
    <xs:attribute ref="xml:lang" use="optional" />
  </xs:restriction>
</xs:complexContent>
</xs:complexType>

```

Common Word Schema

```

<xsd:complexType name="CT_String">
  <xsd:attribute name="val" type="ST_String" use="required">
    <xsd:annotation>
      <xsd:documentation>String Value</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>

<xsd:complexType name="CT_Text">
  <xsd:simpleContent>
    <xsd:extension base="ST_String">
      <xsd:attribute ref="xml:space" use="optional">
        <xsd:annotation>
          <xsd:documentation>Content Contains Significant Whitespace</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

<xsd:simpleType name="ST_String">
  <xsd:annotation>
    <xsd:documentation>String</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string" />
</xsd:simpleType>

<xsd:simpleType name="ST_DecimalNumber">
  <xsd:annotation>
    <xsd:documentation>Decimal Number Value</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:integer" />
</xsd:simpleType>

<xsd:complexType name="CT_DecimalNumber">
  <xsd:attribute name="val" type="ST_DecimalNumber" use="required">

```

```

        <xsd:annotation>
          <xsd:documentation>Decimal Number Value</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    </xsd:complexType>

    <xsd:complexType name="CT_Rel">
      <xsd:attribute ref="r:id" use="required">
        <xsd:annotation>
          <xsd:documentation>Relationship to Part</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    </xsd:complexType>

    <xsd:simpleType name="ST_OnOff">
      <xsd:annotation>
        <xsd:documentation>On/Off Value</xsd:documentation>
      </xsd:annotation>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="true">
          <xsd:annotation>
            <xsd:documentation>True</xsd:documentation>
          </xsd:annotation>
        </xsd:enumeration>
        <xsd:enumeration value="false">
          <xsd:annotation>
            <xsd:documentation>False</xsd:documentation>
          </xsd:annotation>
        </xsd:enumeration>
        <xsd:enumeration value="on">
          <xsd:annotation>
            <xsd:documentation>True</xsd:documentation>
          </xsd:annotation>
        </xsd:enumeration>
        <xsd:enumeration value="off">
          <xsd:annotation>
            <xsd:documentation>False</xsd:documentation>
          </xsd:annotation>
        </xsd:enumeration>
        <xsd:enumeration value="0">
          <xsd:annotation>
            <xsd:documentation>False</xsd:documentation>
          </xsd:annotation>
        </xsd:enumeration>
        <xsd:enumeration value="1">
          <xsd:annotation>
            <xsd:documentation>True</xsd:documentation>
          </xsd:annotation>
        </xsd:enumeration>
      </xsd:restriction>
    </xsd:simpleType>

    <xsd:complexType name="CT_OnOff">
      <xsd:attribute name="val" type="ST_OnOff">
        <xsd:annotation>
          <xsd:documentation>On/Off Value</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    </xsd:complexType>

```

```

        </xsd:annotation>
      </xsd:attribute>
    </xsd:complexType>

    <xsd:simpleType name="ST_DateTime">
      <xsd:annotation>
        <xsd:documentation>Standard Date and Time Storage Format
        </xsd:documentation>
      </xsd:annotation>
      <xsd:restriction base="xsd:dateTime" />
    </xsd:simpleType>

    <xsd:simpleType name="ST_DisplacedByCustomXml">
      <xsd:annotation>
        <xsd:documentation>Location of Custom XML Markup Displacing an
          Annotation</xsd:documentation>
      </xsd:annotation>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="next">
          <xsd:annotation>
            <xsd:documentation>Displaced by Next Custom XML Markup Tag
            </xsd:documentation>
          </xsd:annotation>
        </xsd:enumeration>
        <xsd:enumeration value="prev">
          <xsd:annotation>
            <xsd:documentation>Displaced by Previous Custom XML Markup Tag
            </xsd:documentation>
          </xsd:annotation>
        </xsd:enumeration>
      </xsd:restriction>
    </xsd:simpleType>

    <xsd:complexType name="CT_MarkupRange">
      <xsd:complexContent>
        <xsd:extension base="CT_Markup">
          <xsd:attribute name="displacedByCustomXml"
            type="ST_DisplacedByCustomXml"
            use="optional">
            <xsd:annotation>
              <xsd:documentation>Annotation Marker Relocated For Custom
                XML Markup</xsd:documentation>
            </xsd:annotation>
          </xsd:attribute>
        </xsd:extension>
      </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="CT_Markup">
      <xsd:attribute name="id" type="">
        <xsd:complexType name="CT_Markup">
          <xsd:attribute name="id" type="ST_DecimalNumber"
            use="required">
            <xsd:annotation>
              <xsd:documentation>Annotation Identifier</xsd:documentation>
            </xsd:annotation>

```

```

        </xsd:attribute>
    </xsd:complexType>
    <xsd:annotation>
        <xsd:documentation>Annotation Identifier</xsd:documentation>
    </xsd:annotation>
</xsd:attribute>
</xsd:complexType>

```

Common Excel Schema

```

<xsd:simpleType name="ST_Xstring">
    <xsd:annotation>
        <xsd:documentation>String With Encoded Characters
    </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string" />
</xsd:simpleType>

<xsd:complexType name="CT_Rst">
    <xsd:sequence>
        <xsd:element name="t" type="ST_Xstring" minOccurs="0"
            maxOccurs="1">
            <xsd:annotation>
                <xsd:documentation>Text</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CT_InputCells">
    <xsd:attribute name="r" type="ST_CellRef" use="required">
        <xsd:annotation>
            <xsd:documentation>Reference</xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="deleted" type="xsd:boolean" use="optional"
        default="false">
        <xsd:annotation>
            <xsd:documentation>Deleted</xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="undone" type="xsd:boolean" use="optional"
        default="false">
        <xsd:annotation>
            <xsd:documentation>Undone</xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="val" type="ST_Xstring" use="required">
        <xsd:annotation>
            <xsd:documentation>Value</xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>

```

```

</xsd:attribute>
<xsd:attribute name="numFmtId" type="ST_NumFmtId" use="optional">
  <xsd:annotation>
    <xsd:documentation>Number Format Id</xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
</xsd:complexType>

<xsd:simpleType name="ST_Sqref">
  <xsd:annotation>
    <xsd:documentation>Reference Sequence</xsd:documentation>
  </xsd:annotation>
  <xsd:list itemType="ST_Ref" />
</xsd:simpleType>

<xsd:simpleType name="ST_Ref">
  <xsd:annotation>
    <xsd:documentation>Cell References</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string" />
</xsd:simpleType>

<xsd:simpleType name="ST_CellRef">
  <xsd:annotation>
    <xsd:documentation>Cell Reference</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string" />
</xsd:simpleType>

<xsd:simpleType name="ST_NumFmtId">
  <xsd:annotation>
    <xsd:documentation>Number Format Id</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:unsignedInt" />
</xsd:simpleType>

```

Common PowerPoint Schema

```

<xsd:group name="EG_ExtensionList">
  <xsd:sequence>
    <xsd:element name="ext" type="CT_Extension" minOccurs="0"
maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation>Extension</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:group>

<xsd:complexType name="CT_ExtensionList">
  <xsd:sequence>
    <xsd:group ref="EG_ExtensionList" minOccurs="0" maxOccurs="1" />
  </xsd:sequence>
</xsd:complexType>

```



```

<xsd:complexType name="CT_Extension">
  <xsd:sequence>
    <xsd:any processContents="lax" />
  </xsd:sequence>
  <xsd:attribute name="uri" type="xsd:token">
    <xsd:annotation>
      <xsd:documentation>Uniform Resource
Identifier</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>

<xsd:complexType name="CT_ExtensionListModify">
  <xsd:sequence>
    <xsd:group ref="EG_ExtensionList" minOccurs="0" maxOccurs="1" />
  </xsd:sequence>
  <xsd:attribute name="mod" type="xsd:boolean" use="optional"
default="false">
    <xsd:annotation>
      <xsd:documentation>Modify</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>

<xsd:simpleType name="ST_Name">
  <xsd:annotation>
    <xsd:documentation>Name string</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string"></xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="ST_Index">
  <xsd:annotation>
    <xsd:documentation>Index</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:unsignedInt" />
</xsd:simpleType>

<xsd:complexType name="CT_Point2D">
  <xsd:attribute name="x" type="ST_Coordinate" use="required">
    <xsd:annotation>
      <xsd:documentation>X-Axis Coordinate</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="y" type="ST_Coordinate" use="required">
    <xsd:annotation>
      <xsd:documentation>Y-Axis Coordinate</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>

<xsd:simpleType name="ST_Coordinate">
  <xsd:annotation>
    <xsd:documentation>Coordinate</xsd:documentation>
  </xsd:annotation>

```

```

<xsd:restriction base="xsd:long">
  <xsd:minInclusive value="-27273042329600" />
  <xsd:maxInclusive value="27273042316900" />
</xsd:restriction>
</xsd:simpleType>

```

Metadata Schema

OFFICE 2007.5.1: **Field Codes**

PRODUCT: WORD

```

<xsd:element name="fldSimple" type="CT_SimpleField" minOccurs="0"
maxOccurs="unbounded">
  <xsd:annotation>
    <xsd:documentation>Simple Field</xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:complexType name="CT_SimpleField">
  <xsd:sequence>
    <xsd:element name="fldData" type="CT_Text" minOccurs="0"
maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Custom Field Data</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:group ref="EG_PContent" minOccurs="0" maxOccurs="unbounded"
/>
  </xsd:sequence>
  <xsd:attribute name="instr" type="ST_String" use="required">
    <xsd:annotation>
      <xsd:documentation>Field Codes</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="fldLock" type="ST_OnOff">
    <xsd:annotation>
      <xsd:documentation>Field Should Not Be
Recalculated</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="dirty" type="ST_OnOff">
    <xsd:annotation>
      <xsd:documentation>Field Result Invalidated</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>

---

<xsd:element name="fldChar" type="CT_FldChar">
  <xsd:annotation>
    <xsd:documentation>Complex Field Character</xsd:documentation>
  </xsd:annotation>
</xsd:element>

```

```

<xsd:element name="instrText" type="CT_Text">
  <xsd:annotation>
    <xsd:documentation>Field Code</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:complexType name="CT_FldChar">
  <xsd:choice>
    <xsd:element name="fldData" type="CT_Text" minOccurs="0"
maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Custom Field Data</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="ffData" type="CT_FFData" minOccurs="0"
maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Form Field Properties</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="numberingChange" type="CT_TrackChangeNumbering"
minOccurs="0">
      <xsd:annotation>
        <xsd:documentation>Previous Numbering Field
Properties</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:choice>
  <xsd:attribute name="fldCharType" type="ST_FldCharType"
use="required">
    <xsd:annotation>
      <xsd:documentation>Field Character Type</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="fldLock" type="ST_OnOff">
    <xsd:annotation>
      <xsd:documentation>Field Should Not Be
Recalculated</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="dirty" type="ST_OnOff">
    <xsd:annotation>
      <xsd:documentation>Field Result Invalidated</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>

<xsd:complexType name="CT_Text">
  <xsd:simpleContent>
    <xsd:extension base="ST_String">
      <xsd:attribute ref="xml:space" use="optional">
        <xsd:annotation>
          <xsd:documentation>Content Contains Significant
Whitespace</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

```

```
</xsd:extension>  
</xsd:simpleContent>  
</xsd:complexType>
```

OFFICE 2007.5.1: **END**

OFFICE 2007.5.2: Macros**PRODUCT: WORD**

/[Content_Types].xml:

```

<xs:element name="Types" type="CT_Types" />
<xs:element name="Default" type="CT_Default" />
<xs:element name="Override" type="CT_Override" />

<xs:complexType name="CT_Types">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element ref="Default" />
    <xs:element ref="Override" />
  </xs:choice>
</xs:complexType>

<xs:complexType name="CT_Default">
  <xs:attribute name="Extension" type="ST_Extension" use="required" />
  <xs:attribute name="ContentType" type="ST_ContentType"
use="required" />
</xs:complexType>

<xs:complexType name="CT_Override">
  <xs:attribute name="ContentType" type="ST_ContentType"
use="required" />
  <xs:attribute name="PartName" type="xs:anyURI" use="required" />
</xs:complexType>

```

/word/vbaData.xml:

```

<xsd:element name="vbaSuppData">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="mcDs">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="mcd">
              <xsd:complexType>
                <xsd:attribute name="macroName" type="xsd:string" />
                <xsd:attribute name="name" type="xsd:string" />
                <xsd:attribute name="bEncrypt" type="xsd:integer" />
                <xsd:attribute name="cmg" type="xsd:integer" />
              </xsd:complexType>
            </xsd:element>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

```

PRODUCT: EXCEL

/[Content_Types].xml:

```

<xs:element name="Types" type="CT_Types" />
<xs:element name="Default" type="CT_Default" />

```

```

<xs:element name="Override" type="CT_Override" />

<xs:complexType name="CT_Types">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element ref="Default" />
    <xs:element ref="Override" />
  </xs:choice>
</xs:complexType>

<xs:complexType name="CT_Default">
  <xs:attribute name="Extension" type="ST_Extension" use="required" />
  <xs:attribute name="ContentType" type="ST_ContentType"
use="required" />
</xs:complexType>

<xs:complexType name="CT_Override">
  <xs:attribute name="ContentType" type="ST_ContentType"
use="required" />
  <xs:attribute name="PartName" type="xs:anyURI" use="required" />
</xs:complexType>

```

PRODUCT: POWERPOINT

/[Content_Types].xml:

```

<xs:element name="Types" type="CT_Types" />
<xs:element name="Default" type="CT_Default" />
<xs:element name="Override" type="CT_Override" />

<xs:complexType name="CT_Types">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element ref="Default" />
    <xs:element ref="Override" />
  </xs:choice>
</xs:complexType>

<xs:complexType name="CT_Default">
  <xs:attribute name="Extension" type="ST_Extension" use="required" />
  <xs:attribute name="ContentType" type="ST_ContentType"
use="required" />
</xs:complexType>

<xs:complexType name="CT_Override">
  <xs:attribute name="ContentType" type="ST_ContentType"
use="required" />
  <xs:attribute name="PartName" type="xs:anyURI" use="required" />
</xs:complexType>

```

OFFICE 2007.5.2: **END**

OFFICE 2007.5.3: **Comments**

PRODUCT: WORD

```

<!-- Review comments from document*.xml -->
<xsd:element name="commentRangeStart" type="CT_MarkupRange">
  <xsd:annotation>
    <xsd:documentation>Comment Anchor Range Start</xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:element name="commentRangeEnd" type="CT_MarkupRange">
  <xsd:annotation>
    <xsd:documentation>Comment Anchor Range End</xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:complexType name="CT_R">
  <xsd:element name="commentReference" type="CT_Markup">
    <xsd:annotation>
      <xsd:documentation>Comment Content Reference
Mark</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
</xsd:complexType>

<xsd:element name="r" type="CT_R">
  <xsd:annotation>
    <xsd:documentation>Text Run</xsd:documentation>
  </xsd:annotation>
</xsd:element>

<!-- from comments*.xml -->
<xsd:complexType name="CT_Comments">
  <xsd:sequence>
    <xsd:element name="comment" type="CT_Comment"
      minOccurs="0" maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation>Comment Content</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:element name="comments" type="CT_Comments">
  <xsd:annotation>
    <xsd:documentation>Comments Collection</xsd:documentation>
  </xsd:annotation>
</xsd:element>

<!-- In-text comments in document*.xml -->
<xsd:element name="sdt" type="xsd:time">
  <xsd:annotation>
    <xsd:documentation>Document</xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:complexType name="CT_SdtBlock">

```

```

<xsd:sequence>
  <xsd:element name="sdtPr" type="CT_SdtPr" minOccurs="0"
    maxOccurs="1">
    <xsd:annotation>
      <xsd:documentation>Structured Document Tag Properties
    </xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="sdtContent" type="CT_SdtContentBlock"
    minOccurs="0" maxOccurs="1">
    <xsd:annotation>
      <xsd:documentation>Block-Level Structured Document Tag Content
    </xsd:documentation>
    </xsd:annotation>
  </xsd:element>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CT_SdtPr">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="alias" type="CT_StringWordComment"
      minOccurs="0">
      <xsd:annotation>
        <xsd:documentation>Friendly Name</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="placeholder" type="CT_Placeholder"
      minOccurs="0">
      <xsd:annotation>
        <xsd:documentation>Structured Document Tag Placeholder Text
      </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="dataBinding">
      <xsd:annotation>
        <xsd:documentation>XML Mapping</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="id" type="CT_DecimalNumber"
      minOccurs="0">
      <xsd:annotation>
        <xsd:documentation>Unique ID</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:choice>
</xsd:complexType>

<xsd:complexType name="CT_Placeholder">
  <xsd:sequence>
    <xsd:element name="docPart" type="CT_String">
      <xsd:annotation>
        <xsd:documentation>Document Part Reference</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

```



```

<xsd:complexType name="CT_StringWordComment">
  <xsd:attribute name="val" type="xsd:string" fixed="Comments">
    <xsd:annotation>
      <xsd:documentation>String Value</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>

<xsd:complexType name="CT_SdtContentBlock">
  <xsd:sequence>
    <xsd:element name="p" type="CT_P" minOccurs="0"
      maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation>Paragraph</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CT_P">
  <xsd:sequence>
    <xsd:element name="r" type="CT_R">
      <xsd:annotation>
        <xsd:documentation>Text Run</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CT_R">
  <xsd:sequence>
    <xsd:element name="t" type="CT_Text">
      <xsd:annotation>
        <xsd:documentation>Text</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

```

PRODUCT: EXCEL

```

<xsd:element name="comments" type="CT_Comments">
  <xsd:annotation>
    <xsd:documentation>Comments</xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:complexType name="CT_Comments">
  <xsd:sequence>
    <xsd:element name="authors" type="CT_Authors"
      minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Authors</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>

```

```

<xsd:element name="commentList" type="CT_CommentList"
  minOccurs="1" maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>List of Comments</xsd:documentation>
  </xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CT_Authors">
  <xsd:sequence>
    <xsd:element name="author" type="ST_Xstring"
      minOccurs="0" maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation>Author</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CT_CommentList">
  <xsd:sequence>
    <xsd:element name="comment" type="CT_Comment"
      minOccurs="0" maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation>Comment</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CT_Comment">
  <xsd:sequence>
    <xsd:element name="text" type="CT_Rst" minOccurs="1"
      maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Comment Text</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="ST_Xstring">
  <xsd:annotation>
    <xsd:documentation>String With Encoded Characters
  </xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string" />
</xsd:simpleType>

<xsd:complexType name="CT_Rst">
  <xsd:sequence>
    <xsd:element name="t" type="ST_Xstring" minOccurs="0"
      maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Text</xsd:documentation>

```

```

        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>

```

PRODUCT: POWERPOINT

```

<!-- from commentAuthors.xml -->
<xsd:complexType name="CT_CommentAuthor">
  <xsd:sequence>
    <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0"
maxOccurs="1" />
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:unsignedInt" use="required">
    <xsd:annotation>
      <xsd:documentation>Comment Author ID</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="name" type="ST_Name" use="required">
    <xsd:annotation>
      <xsd:documentation>Comment Author Name</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="initials" type="ST_Name" use="required">
    <xsd:annotation>
      <xsd:documentation>Comment Author Initials</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="lastIdx" type="xsd:unsignedInt" use="required">
    <xsd:annotation>
      <xsd:documentation>Index of Comment Author's last
comment</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="clrIdx" type="xsd:unsignedInt" use="required">
    <xsd:annotation>
      <xsd:documentation>Comment Author Color
Index</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>

<xsd:complexType name="CT_CommentAuthorList">
  <xsd:sequence>
    <xsd:element name="cmAuthor" type="CT_CommentAuthor" minOccurs="0"
maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation>Comment Author</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:element name="cmAuthorLst" type="CT_CommentAuthorList">
  <xsd:annotation>
    <xsd:documentation>List of Comment Authors</xsd:documentation>
  </xsd:annotation>

```

```

</xsd:element>

<!-- from comment*.xml -->
<xsd:complexType name="CT_Comment">
  <xsd:sequence>
    <xsd:element name="pos" type="a:CT_Point2D" minOccurs="1"
maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Comment Position</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="text" type="xsd:string" minOccurs="1"
maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Comment's Text Content</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="extLst" type="CT_ExtensionListModify"
minOccurs="0" maxOccurs="1" />
  </xsd:sequence>
  <xsd:attribute name="authorId" type="xsd:unsignedInt"
use="required">
    <xsd:annotation>
      <xsd:documentation>Comment Author ID</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="dt" type="xsd:dateTime" use="optional">
    <xsd:annotation>
      <xsd:documentation>Comment Date/Time</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="idx" type="ST_Index" use="required">
    <xsd:annotation>
      <xsd:documentation>Comment Index</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>

<xsd:complexType name="CT_CommentList">
  <xsd:sequence>
    <xsd:element name="cm" type="CT_Comment" minOccurs="0"
maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation>Comment</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:element name="cmLst" type="CT_CommentList">
  <xsd:annotation>
    <xsd:documentation>Comment List</xsd:documentation>
  </xsd:annotation>
</xsd:element>

<!-- from notesSlide*.xml -->

```

```

<xsd:element name="notes" type="CT_NotesSlide">
  <xsd:annotation>
    <xsd:documentation>Notes Slide</xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:complexType name="CT_NotesSlide">
  <xsd:sequence minOccurs="1" maxOccurs="1">
    <xsd:element name="cSld" type="CT_CommonSlideData"
      minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Common slide data for notes slides
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CT_CommonSlideData">
  <xsd:sequence>
    <xsd:element name="spTree" type="CT_GroupShape"
      minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Shape Tree</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CT_GroupShape">
  <xsd:sequence>
    <xsd:element name="sp" type="CT_Shape">
      <xsd:annotation>
        <xsd:documentation>Shape</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CT_Shape">
  <xsd:sequence>
    <xsd:element name="txBody" type="CT_TextBody"
      minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Shape Text Body</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CT_TextBody">
  <xsd:sequence>
    <xsd:element name="p" type="CT_TextParagraph"
      minOccurs="1" maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation>Text Paragraphs</xsd:documentation>

```

```

        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="CT_TextParagraph">
    <xsd:sequence>
      <xsd:element name="r" type="CT_RegularTextRun">
        <xsd:annotation>
          <xsd:documentation>Text Run</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="CT_RegularTextRun">
    <xsd:sequence>
      <xsd:element name="t" type="xsd:string" minOccurs="1"
        maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>Text String</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>

```

OFFICE 2007.5.3: **END**

OFFICE 2007.5.4: **Linked Objects** **PRODUCT: WORD**

```

<!-- Word OLE Objects in document*.xml -->
<xsd:element name="object" type="CT_Object">
  <xsd:annotation>
    <xsd:documentation>Inline Embedded Object</xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:complexType name="CT_Object">
  <xsd:complexContent>
    <xsd:extension base="CT_PictureBase">
      <xsd:sequence>
        <xsd:element name="control" type="CT_Control"
          minOccurs="0">
          <xsd:annotation>
            <xsd:documentation>Inline Embedded Control
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
      <xsd:attribute name="dxaOrig" type="ST_TwipsMeasure"
        use="optional">
        <xsd:annotation>

```

```

        <xsd:documentation>Original Image Width</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="dyaOrig" type="ST_TwipsMeasure"
      use="optional">
      <xsd:annotation>
        <xsd:documentation>Original Image Height</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  </xsd:extension>
</xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="CT_PictureBase">
  <xsd:sequence maxOccurs="unbounded">
    <xsd:any processContents="lax" namespace="urn:schemas-microsoft-
com:vml"
      minOccurs="0" />
    <xsd:any processContents="lax"
      namespace="urn:schemas-microsoft-com:office:office"
minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CT_Control">
  <xsd:attribute name="name" type="ST_String" use="optional">
    <xsd:annotation>
      <xsd:documentation>Unique Name for Embedded Control
    </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="shapeid" type="ST_String" use="optional">
    <xsd:annotation>
      <xsd:documentation>Associated VML Data Reference
    </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute ref="r:id" use="optional">
    <xsd:annotation>
      <xsd:documentation>Embedded Control Properties Relationship
      Reference</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>

```

PRODUCT: EXCEL

```

<xsd:element name="oleObjects" type="CT_OleObjects"
  minOccurs="0" maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>OLE Objects</xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:complexType name="CT_OleObjects">
  <xsd:sequence>

```

```

        <xsd:element name="oleObject" type="CT_OleObject"
          minOccurs="1" maxOccurs="unbounded">
          <xsd:annotation>
            <xsd:documentation>OLE Object</xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="CT_OleObject">
      <xsd:attribute name="progId" type="xsd:string" use="optional">
        <xsd:annotation>
          <xsd:documentation>OLE ProgId</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="link" type="ST_Xstring" use="optional">
        <xsd:annotation>
          <xsd:documentation>OLE Link Moniker</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="autoLoad" type="xsd:boolean" use="optional"
        default="false">
        <xsd:annotation>
          <xsd:documentation>Auto Load</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="shapeId" type="xsd:unsignedInt"
        use="required">
        <xsd:annotation>
          <xsd:documentation>Shape Id</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute ref="r:id" use="optional">
        <xsd:annotation>
          <xsd:documentation>Relationship Id</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    </xsd:complexType>

```

PRODUCT: POWERPOINT

```

    <xsd:element name="oleObj" type="CT_OleObject">
      <xsd:annotation>
        <xsd:documentation>Global Element for OLE Objects and Controls
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>

    <xsd:complexType name="CT_OleObject">
      <xsd:choice minOccurs="1" maxOccurs="1">
        <xsd:element name="embed" type="CT_OleObjectEmbed">
          <xsd:annotation>
            <xsd:documentation>Embedded OLE Object or Control
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:choice>
    </xsd:complexType>

```



```

        <xsd:element name="link" type="CT_OleObjectLink">
          <xsd:annotation>
            <xsd:documentation>Linked OLE Object or Control
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:choice>
    <xsd:attributeGroup ref="AG_Ole" />
    <xsd:attribute name="progId" type="xsd:string" use="optional">
      <xsd:annotation>
        <xsd:documentation>OLE ProgID</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  </xsd:complexType>

  <xsd:complexType name="CT_OleObjectEmbed">
    <xsd:sequence>
      <xsd:element name="extLst" type="CT_ExtensionList"
        minOccurs="0" maxOccurs="1" />
    </xsd:sequence>
    <xsd:attribute name="followColorScheme"
      type="ST_OleObjectFollowColorScheme"
      use="optional" default="none">
      <xsd:annotation>
        <xsd:documentation>Color Scheme Properties for OLE Object
        </xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  </xsd:complexType>

  <xsd:simpleType name="ST_OleObjectFollowColorScheme">
    <xsd:annotation>
      <xsd:documentation>OLE Object to Follow Color Scheme
      </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:token">
      <xsd:enumeration value="none">
        <xsd:annotation>
          <xsd:documentation>None</xsd:documentation>
        </xsd:annotation>
      </xsd:enumeration>
      <xsd:enumeration value="full">
        <xsd:annotation>
          <xsd:documentation>Full</xsd:documentation>
        </xsd:annotation>
      </xsd:enumeration>
      <xsd:enumeration value="textAndBackground">
        <xsd:annotation>
          <xsd:documentation>Text and Background</xsd:documentation>
        </xsd:annotation>
      </xsd:enumeration>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:complexType name="CT_OleObjectLink">
    <xsd:sequence>

```

```

        <xsd:element name="extLst" type="CT_ExtensionList"
            minOccurs="0" maxOccurs="1" />
    </xsd:sequence>
    <xsd:attribute name="updateAutomatic" type="xsd:boolean"
        use="optional" default="false">
        <xsd:annotation>
            <xsd:documentation>Update Linked OLE Objects Automatically
        </xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
</xsd:complexType>

<xsd:complexType name="CT_Extension">
    <xsd:sequence>
        <xsd:any processContents="lax" />
    </xsd:sequence>
    <xsd:attribute name="uri" type="xsd:token">
        <xsd:annotation>
            <xsd:documentation>Uniform Resource Identifier
        </xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
</xsd:complexType>

<xsd:group name="EG_ExtensionList">
    <xsd:sequence>
        <xsd:element name="ext" type="CT_Extension" minOccurs="0"
            maxOccurs="unbounded">
            <xsd:annotation>
                <xsd:documentation>Extension</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
    </xsd:sequence>
</xsd:group>

<xsd:complexType name="CT_ExtensionList">
    <xsd:sequence>
        <xsd:group ref="EG_ExtensionList" minOccurs="0"
            maxOccurs="1" />
    </xsd:sequence>
</xsd:complexType>

<xsd:attributeGroup name="AG_Ole">
    <xsd:attribute name="spid" type="a:ST_ShapeID" use="required">
        <xsd:annotation>
            <xsd:documentation>OLE Object Shape ID</xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="name" type="xsd:string" use="optional"
        default="">
        <xsd:annotation>
            <xsd:documentation>OLE Object Name</xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="showAsIcon" type="xsd:boolean"
        use="optional" default="false">

```

```

        <xsd:annotation>
          <xsd:documentation>Show OLE Object As Icon</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute ref="r:id" use="optional">
        <xsd:annotation>
          <xsd:documentation>Relationship ID</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="imgW" type="a:ST_PositiveCoordinate32"
        use="optional">
        <xsd:annotation>
          <xsd:documentation>Image Width</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="imgH" type="a:ST_PositiveCoordinate32"
        use="optional">
        <xsd:annotation>
          <xsd:documentation>Image Height</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    </xsd:attributeGroup>

```

OFFICE 2007.5.4: **END**

OFFICE 2007.5.5: **Template Name**
PRODUCT: WORD

```

<!-- from app*.xml -->
<xsd:element name="Properties" type="CT_Properties">
  <xsd:annotation>
    <xsd:documentation>Application Specific File Properties
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:complexType name="CT_Properties">
  <xsd:sequence>
    <xsd:element name="Template" minOccurs="0" maxOccurs="1"
      type="xsd:string">
      <xsd:annotation>
        <xsd:documentation>Name of Document Template
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<!-- from settings*.xml -->
<xsd:element name="settings" type="CT_Settings">
  <xsd:annotation>
    <xsd:documentation>Document Settings</xsd:documentation>
  </xsd:annotation>

```

```

</xsd:element>

<xsd:complexType name="CT_Settings">
  <xsd:sequence>
    <xsd:element name="attachedTemplate" type="CT_Rel" minOccurs="0">
      <xsd:annotation>
        <xsd:documentation>Attached Document
Template</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

```

OFFICE 2007.5.5: **END**

OFFICE 2007.5.6: **Scenarios**
PRODUCT: EXCEL

```

<xsd:element name="worksheet" type="CT_Worksheet">
  <xsd:annotation>
    <xsd:documentation>Worksheet</xsd:documentation>

    </xsd:annotation>
  </xsd:element>

<xsd:complexType name="CT_Worksheet">
  <xsd:sequence>
    <xsd:element name="scenarios" type="CT_Scenarios"
      minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Scenarios</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CT_Scenarios">
  <xsd:sequence>
    <xsd:element name="scenario" type="CT_Scenario"
      minOccurs="1" maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation>Scenario</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
  <xsd:attribute name="current" type="xsd:unsignedInt"
    use="optional">
    <xsd:annotation>
      <xsd:documentation>Current Scenario</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="show" type="xsd:unsignedInt" use="optional">

```

```

        <xsd:annotation>
          <xsd:documentation>Last Shown Scenario</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="sqref" type="ST_Sqref" use="optional">
        <xsd:annotation>
          <xsd:documentation>Sequence of References</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    </xsd:complexType>

    <xsd:complexType name="CT_Scenario">
      <xsd:sequence>
        <xsd:element name="inputCells" type="CT_InputCells"
          minOccurs="1" maxOccurs="unbounded">
          <xsd:annotation>
            <xsd:documentation>Input Cells</xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
      <xsd:attribute name="name" type="ST_Xstring" use="required">
        <xsd:annotation>
          <xsd:documentation>Scenario Name</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="locked" type="xsd:boolean" use="optional"
        default="false">
        <xsd:annotation>
          <xsd:documentation>Scenario Locked</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="hidden" type="xsd:boolean" use="optional"
        default="false">
        <xsd:annotation>
          <xsd:documentation>Hidden Scenario</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="count" type="xsd:unsignedInt"
        use="optional">
        <xsd:annotation>
          <xsd:documentation>Changing Cell Count</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="user" type="ST_Xstring" use="optional">
        <xsd:annotation>
          <xsd:documentation>User Name</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="comment" type="ST_Xstring" use="optional">
        <xsd:annotation>
          <xsd:documentation>Scenario Comment</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    </xsd:complexType>

```

OFFICE 2007.5.6: **END**

OFFICE 2007.5.7: Hyperlinks**PRODUCT: WORD**

```

<xsd:element name="hyperlink" type="CT_Hyperlink">
  <xsd:annotation>
    <xsd:documentation>Hyperlink</xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:complexType name="CT_Hyperlink">
  <xsd:group ref="EG_PContent" minOccurs="0" maxOccurs="unbounded" />
  <xsd:attribute name="tgtFrame" type="ST_String" use="optional">
    <xsd:annotation>
      <xsd:documentation>Hyperlink Target Frame</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="tooltip" type="ST_String" use="optional">
    <xsd:annotation>
      <xsd:documentation>Associated String</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="docLocation" type="ST_String" use="optional">
    <xsd:annotation>
      <xsd:documentation>Location in Target
Document</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="history" type="ST_OnOff" use="optional">
    <xsd:annotation>
      <xsd:documentation>Add To Viewed Hyperlinks</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="anchor" type="ST_String" use="optional">
    <xsd:annotation>
      <xsd:documentation>Hyperlink Anchor</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute ref="r:id">
    <xsd:annotation>
      <xsd:documentation>Hyperlink Target</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>

```

PRODUCT: EXCEL

```

<xsd:element name="hyperlinks" type="CT_Hyperlinks" minOccurs="0"
maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>Hyperlinks</xsd:documentation>
  </xsd:annotation>
</xsd:element>

```

```

<xsd:complexType name="CT_Hyperlinks">
  <xsd:sequence>
    <xsd:element name="hyperlink" type="CT_Hyperlink" minOccurs="1"
maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation>Hyperlink</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="CT_Hyperlink">
  <xsd:attribute name="ref" type="ST_Ref" use="required">
    <xsd:annotation>
      <xsd:documentation>Reference</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute ref="r:id" use="optional">
    <xsd:annotation>
      <xsd:documentation>Relationship Id</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="location" type="ST_Xstring" use="optional">
    <xsd:annotation>
      <xsd:documentation>Location</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="tooltip" type="ST_Xstring" use="optional">
    <xsd:annotation>
      <xsd:documentation>Tool Tip</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="display" type="ST_Xstring" use="optional">
    <xsd:annotation>
      <xsd:documentation>Display String</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>

```

PRODUCT: POWERPOINT

```

<xsd:element name="hlinkClick" type="CT_Hyperlink" minOccurs="0"
maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>Click Hyperlink</xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:complexType name="CT_Hyperlink">
  <xsd:sequence>
    <xsd:element name="snd" type="CT_EmbeddedWAVAudioFile"
minOccurs="0" maxOccurs="1">
      <xsd:annotation>

```

```

        <xsd:documentation>Hyperlink Sound</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="extLst" type="CT_OfficeArtExtensionList"
minOccurs="0" maxOccurs="1"></xsd:element>
  </xsd:sequence>
  <xsd:attribute ref="r:id" use="optional">
    <xsd:annotation>
      <xsd:documentation>Drawing Object Hyperlink
Target</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="invalidUrl" type="xsd:string" use="optional"
default="">
    <xsd:annotation>
      <xsd:documentation>Invalid URL</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="action" type="xsd:string" use="optional"
default="">
    <xsd:annotation>
      <xsd:documentation>Action Setting</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="tgtFrame" type="xsd:string" use="optional"
default="">
    <xsd:annotation>
      <xsd:documentation>Target Frame</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="tooltip" type="xsd:string" use="optional"
default="">
    <xsd:annotation>
      <xsd:documentation>Hyperlink Tooltip</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="history" type="xsd:boolean" use="optional"
default="true">
    <xsd:annotation>
      <xsd:documentation>Add Hyperlink to Page
History</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="highlightClick" type="xsd:boolean"
use="optional" default="false">
    <xsd:annotation>
      <xsd:documentation>Highlight Click</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="endSnd" type="xsd:boolean" use="optional"
default="false">
    <xsd:annotation>
      <xsd:documentation>End Sounds</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>

```


OFFICE 2007.5.7:	END
------------------	------------

OFFICE 2007.5.8:	Core Properties PRODUCT: WORD, EXCEL, AND POWERPOINT OOXML uses the Dublin Core ⁷ Metadata Element Set. OFFICE 2007.5.8: END
------------------	---

OFFICE 2007.5.9:	Custom Properties PRODUCT: WORD, EXCEL, AND POWERPOINT <pre> <xsd:element name="Properties" type="CT_Properties"> <xsd:annotation> <xsd:documentation>Custom File Properties</xsd:documentation> </xsd:annotation> </xsd:element> <xsd:complexType name="CT_Properties"> <xsd:sequence> <xsd:element name="property" minOccurs="0" maxOccurs="unbounded" type="CT_Property"> <xsd:annotation> <xsd:documentation>Custom File Property</xsd:documentation> </xsd:annotation> </xsd:element> </xsd:sequence> </xsd:complexType> <xsd:complexType name="CT_Property"> <xsd:choice minOccurs="1" maxOccurs="1"> <xsd:element ref="vt:lpwstr"> <xsd:annotation> <xsd:documentation>LPWSTR</xsd:documentation> </xsd:annotation> </xsd:element> <xsd:element ref="vt:filetime"> <xsd:annotation> <xsd:documentation>File Time</xsd:documentation> </xsd:annotation> </xsd:element> <xsd:element ref="vt:bool"> <xsd:annotation> <xsd:documentation>Boolean</xsd:documentation> </xsd:annotation> </xsd:element> <!-- For the complete list of schema elements see shared- documentPropertiesCustom.xsd --> </pre>
------------------	--

⁷ Dublin Core is a registered trademark of the Dublin Core Metadata Initiative

```

        </xsd:choice>
        <xsd:attribute name="fmtid" use="required" type="vt:ST_Clsid">
          <xsd:annotation>
            <xsd:documentation>Format ID</xsd:documentation>
          </xsd:annotation>
        </xsd:attribute>
        <xsd:attribute name="pid" use="required" type="xsd:int">
          <xsd:annotation>
            <xsd:documentation>Property ID</xsd:documentation>
          </xsd:annotation>
        </xsd:attribute>
        <xsd:attribute name="name" use="optional" type="xsd:string">
          <xsd:annotation>
            <xsd:documentation>Custom File Property
Name</xsd:documentation>
          </xsd:annotation>
        </xsd:attribute>
        <xsd:attribute name="linkTarget" use="optional" type="xsd:string">
          <xsd:annotation>
            <xsd:documentation>Bookmark Link Target</xsd:documentation>
          </xsd:annotation>
        </xsd:attribute>
      </xsd:complexType>
    </xsd:schema>

```

OFFICE 2007.5.9: **END**

OFFICE 2007.5.10: **Extended Properties** **PRODUCT: WORD, EXCEL, AND POWERPOINT**

```

    <xsd:element name="Properties" type="CT_Properties">
      <xsd:annotation>
        <xsd:documentation>Application Specific File
Properties</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:complexType name="CT_Properties">
      <xsd:all>
        <xsd:element name="Template" minOccurs="0" maxOccurs="1"
type="xsd:string">
          <xsd:annotation>
            <xsd:documentation>Name of Document
Template</xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="Manager" minOccurs="0" maxOccurs="1"
type="xsd:string">
          <xsd:annotation>
            <xsd:documentation>Name of Manager</xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:all>
    </xsd:complexType>

```

```

        <xsd:element name="Company" minOccurs="0" maxOccurs="1"
type="xsd:string">
        <xsd:annotation>
        <xsd:documentation>Name of Company</xsd:documentation>
        </xsd:annotation>
        </xsd:element>
        <xsd:element name="Pages" minOccurs="0" maxOccurs="1"
type="xsd:int">
        <xsd:annotation>
        <xsd:documentation>Total Number of Pages</xsd:documentation>
        </xsd:annotation>
        </xsd:element>
        <xsd:element name="Words" minOccurs="0" maxOccurs="1"
type="xsd:int">
        <xsd:annotation>
        <xsd:documentation>Word Count</xsd:documentation>
        </xsd:annotation>
        </xsd:element>
        <xsd:element name="Characters" minOccurs="0" maxOccurs="1"
type="xsd:int">
        <xsd:annotation>
        <xsd:documentation>Total Number of
Characters</xsd:documentation>
        </xsd:annotation>
        </xsd:element>
        <xsd:element name="PresentationFormat" minOccurs="0" maxOccurs="1"
type="xsd:string">
        <xsd:annotation>
        <xsd:documentation>Intended Format of
Presentation</xsd:documentation>
        </xsd:annotation>
        </xsd:element>

        <!-- For the complete list of schema elements see shared-
documentPropertiesExtended.xsd -->

        </xsd:all>
    </xsd:complexType>

```

OFFICE 2007.5.10: **END**

OFFICE 2007.5.11: **Footnotes and Endnotes**

PRODUCT: WORD

```

<!-- from document*.xml -->
<xsd:group name="EG_RunInnerContent">
    <xsd:choice>
        <xsd:element name="footnoteReference" type="CT_FtnEdnRef">
            <xsd:annotation>
            <xsd:documentation>Footnote Reference</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="endnoteReference" type="CT_FtnEdnRef">

```

```

        <xsd:annotation>
            <xsd:documentation>Endnote Reference</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
</xsd:choice>
</xsd:group>

<xsd:complexType name="CT_R">
    <xsd:group ref="EG_RunInnerContent" minOccurs="0"
        maxOccurs="unbounded" />
</xsd:complexType>

<xsd:element name="r" type="CT_R">
    <xsd:annotation>
        <xsd:documentation>Text Run</xsd:documentation>
    </xsd:annotation>
</xsd:element>

<xsd:complexType name="CT_FtnEdnRef">
    <xsd:attribute name="customMarkFollows" type="ST_OnOff"
        use="optional">
        <xsd:annotation>
            <xsd:documentation>Suppress Footnote/Endnote Reference Mark
        </xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="id" use="required" type="ST_DecimalNumber">
        <xsd:annotation>
            <xsd:documentation>Footnote/Endnote ID Reference
        </xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
</xsd:complexType>

<!-- from footnotes*.xml and endnotes*.xml -->
<xsd:element name="footnotes" type="CT_Footnotes">
    <xsd:annotation>
        <xsd:documentation>Document Footnotes</xsd:documentation>
    </xsd:annotation>
</xsd:element>

<xsd:complexType name="CT_Footnotes">
    <xsd:sequence maxOccurs="unbounded">
        <xsd:element name="footnote" type="CT_FtnEdn"
            minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Footnote Content</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>

<xsd:element name="endnotes" type="CT_Endnotes">
    <xsd:annotation>
        <xsd:documentation>Document Endnotes</xsd:documentation>
    </xsd:annotation>

```

```

</xsd:element>

<xsd:complexType name="CT_Endnotes">
  <xsd:sequence maxOccurs="unbounded">
    <xsd:element name="endnote" type="CT_FtnEdn"
      minOccurs="0">
      <xsd:annotation>
        <xsd:documentation>Endnote Content</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CT_FtnEdn">
  <xsd:attribute name="type" type="ST_FtnEdn" use="optional">
    <xsd:annotation>
      <xsd:documentation>Footnote/Endnote Type</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="id" type="ST_DecimalNumber" use="required">
    <xsd:annotation>
      <xsd:documentation>Footnote/Endnote ID</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>

<xsd:simpleType name="ST_FtnEdn">
  <xsd:annotation>
    <xsd:documentation>Footnote or Endnote Type</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="normal">
      <xsd:annotation>
        <xsd:documentation>Normal Footnote/Endnote</xsd:documentation>
      </xsd:annotation>
    </xsd:enumeration>
    <xsd:enumeration value="separator">
      <xsd:annotation>
        <xsd:documentation>Separator</xsd:documentation>
      </xsd:annotation>
    </xsd:enumeration>
    <xsd:enumeration value="continuationSeparator">
      <xsd:annotation>
        <xsd:documentation>Continuation Separator</xsd:documentation>
      </xsd:annotation>
    </xsd:enumeration>
    <xsd:enumeration value="continuationNotice">
      <xsd:annotation>
        <xsd:documentation>Continuation Notice Separator</xsd:documentation>
      </xsd:annotation>
    </xsd:enumeration>
  </xsd:restriction>
</xsd:simpleType>

```

OFFICE 2007.5.11: **END**OFFICE 2007.5.12: **Headers and Footers****PRODUCT: WORD**

```

<!-- header*.xml files -->
<xsd:element name="hdr">
  <xsd:annotation>
    <xsd:documentation>Header</xsd:documentation>
  </xsd:annotation>
</xsd:element>

  <!-- footer*.xml files -->
<xsd:element name="ftr">
  <xsd:annotation>
    <xsd:documentation>Footer</xsd:documentation>
  </xsd:annotation>
</xsd:element>

<!-- document*.xml files -->
<xsd:element name="sectPr" type="CT_SectPr">
  <xsd:annotation>
    <xsd:documentation>Document Final Section Properties
  </xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:complexType name="CT_SectPr">
  <xsd:sequence>
    <xsd:group ref="EG_HdrFtrReferences" minOccurs="0"
      maxOccurs="6" />
    <xsd:group ref="EG_SectPrContents" minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>

<xsd:group name="EG_SectPrContents">
  <xsd:sequence>
    <xsd:element name="titlePg" type="CT_OnOff" minOccurs="0">
      <xsd:annotation>
        <xsd:documentation>Different First Page Headers and Footers
      </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:group>

<xsd:group name="EG_HdrFtrReferences">
  <xsd:choice>
    <xsd:element name="headerReference" type="CT_HdrFtrRef"
      minOccurs="0">
      <xsd:annotation>
        <xsd:documentation>Header Reference</xsd:documentation>

```

```

        </xsd:annotation>
      </xsd:element>
      <xsd:element name="footerReference" type="CT_HdrFtrRef"
        minOccurs="0">
        <xsd:annotation>
          <xsd:documentation>Footer Reference</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:choice>
  </xsd:group>

  <xsd:complexType name="CT_HdrFtrRef">
    <xsd:complexContent>
      <xsd:extension base="CT_Rel">
        <xsd:attribute name="type" type="ST_HdrFtr" use="required">
          <xsd:annotation>
            <xsd:documentation>Header or Footer Type</xsd:documentation>
          </xsd:annotation>
        </xsd:attribute>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:simpleType name="ST_HdrFtr">
    <xsd:annotation>
      <xsd:documentation>Header or Footer Type</xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="even">
        <xsd:annotation>
          <xsd:documentation>Even Numbered Pages Only
        </xsd:documentation>
        </xsd:annotation>
      </xsd:enumeration>
      <xsd:enumeration value="default">
        <xsd:annotation>
          <xsd:documentation>Default Header or Footer
        </xsd:documentation>
        </xsd:annotation>
      </xsd:enumeration>
      <xsd:enumeration value="first">
        <xsd:annotation>
          <xsd:documentation>First Page Only</xsd:documentation>
        </xsd:annotation>
      </xsd:enumeration>
    </xsd:restriction>
  </xsd:simpleType>

  <!-- settings*.xml files -->
  <xsd:element name="settings" type="CT_Settings">
    <xsd:annotation>
      <xsd:documentation>Document Settings</xsd:documentation>
    </xsd:annotation>
  </xsd:element>

  <xsd:complexType name="CT_Settings">

```

```

<xsd:sequence>
  <xsd:element name="evenAndOddHeaders" type="CT_OnOff"
minOccurs="0">
    <xsd:annotation>
      <xsd:documentation>Different Even/Odd Page Headers and
Footers</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
</xsd:sequence>
</xsd:complexType>

```

PRODUCT: EXCEL

```

<xsd:element name="headerFooter" type="CT_HeaderFooter"
minOccurs="0" maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>Header Footer Settings</xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:complexType name="CT_HeaderFooter">
  <xsd:sequence>
    <xsd:element name="oddHeader" type="ST_Xstring"
minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Odd Header</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="oddFooter" type="ST_Xstring"
minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Odd Page Footer</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="evenHeader" type="ST_Xstring"
minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Even Page Header</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="evenFooter" type="ST_Xstring"
minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Even Page Footer</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="firstHeader" type="ST_Xstring"
minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>First Page Header</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="firstFooter" type="ST_Xstring"
minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>First Page Footer</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

```



```

        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
    <xsd:attribute name="differentOddEven" type="xsd:boolean"
      default="false">
      <xsd:annotation>
        <xsd:documentation>Different Odd Even Header Footer
      </xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="differentFirst" type="xsd:boolean"
      default="false">
      <xsd:annotation>
        <xsd:documentation>Different First Page</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="scaleWithDoc" type="xsd:boolean"
      default="true">
      <xsd:annotation>
        <xsd:documentation>Scale Header & Footer With Document
      </xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="alignWithMargins" type="xsd:boolean"
      default="true">
      <xsd:annotation>
        <xsd:documentation>Align Margins</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  </xsd:complexType>

```

OFFICE 2007.5.12: **END**

OFFICE 2007.5.13: **Hidden Data**
PRODUCT: WORD

```

    <xsd:element name="rPr" type="CT_RPrBase">
      <xsd:annotation>
        <xsd:documentation>Run Properties</xsd:documentation>
      </xsd:annotation>
    </xsd:element>

    <xsd:complexType name="CT_RPrBase">
      <xsd:sequence>
        <xsd:element name="vanish" type="CT_OnOff" minOccurs="0">
          <xsd:annotation>
            <xsd:documentation>Hidden Text</xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="webHidden" type="CT_OnOff"
          minOccurs="0">
          <xsd:annotation>
            <xsd:documentation>Web Hidden Text</xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>

```

```

        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>

```

PRODUCT: EXCEL

```

<xsd:element name="worksheet" type="CT_Worksheet">
  <xsd:annotation>
    <xsd:documentation>Worksheet</xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:complexType name="CT_Worksheet">
  <xsd:sequence>
    <xsd:element name="cols" type="CT_Cols" minOccurs="0"
      maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation>Column Information</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="sheetData" type="CT_SheetData"
      minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Sheet Data</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="scenarios" type="CT_Scenarios"
      minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Scenarios</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<!-- hidden columns in sheet*.xml -->
<xsd:complexType name="CT_Cols">
  <xsd:sequence>
    <xsd:element name="col" type="CT_Col" minOccurs="1"
      maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation>Column Width & Formatting</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CT_Col">
  <xsd:attribute name="width" type="xsd:double" use="optional">
    <xsd:annotation>
      <xsd:documentation>Column Width</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="hidden" type="xsd:boolean" use="optional"

```

```

        default="false">
        <xsd:annotation>
            <xsd:documentation>Hidden Columns</xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="collapsed" type="xsd:boolean"
        use="optional" default="false">
        <xsd:annotation>
            <xsd:documentation>Collapsed</xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="style" type="xsd:unsignedInt" use="optional"
default="0">
        <xsd:annotation>
            <xsd:documentation>Style</xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
</xsd:complexType>

<!-- Cell Hidden on Lock in sheet*.xml -->
<xsd:complexType name="CT_SheetData">
    <xsd:sequence>
        <xsd:element name="row" type="CT_Row" minOccurs="0"
            maxOccurs="unbounded">
            <xsd:annotation>
                <xsd:documentation>Row</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CT_Row">
    <xsd:sequence>
        <xsd:element name="c" type="CT_Cell" minOccurs="0"
            maxOccurs="unbounded">
            <xsd:annotation>
                <xsd:documentation>Cell</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
    </xsd:sequence>
    <!-- Cell Hidden on lock indicated be its style index in styles xml
-->
    <xsd:attribute name="s" type="xsd:unsignedInt" use="optional"
        default="0">
        <xsd:annotation>
            <xsd:documentation>Style Index</xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
    <!-- Cell Hidden on a Collapsed Group in sheet*.xml -->
    <!-- Row Height of 0 in sheet*.xml -->
    <!-- Hidden by filter action in sheet*.xml -->
    <xsd:attribute name="hidden" type="xsd:boolean" use="optional"
        default="false">
        <xsd:annotation>
            <xsd:documentation>Hidden</xsd:documentation>
        </xsd:annotation>

```

```

</xsd:attribute>
<xsd:attribute name="customHeight" type="xsd:boolean"
  use="optional" default="false">
  <xsd:annotation>
    <xsd:documentation>Custom Height</xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
<!-- Cell Hidden on a Collapsed Group in sheet*.xml ver 2 -->
<xsd:attribute name="collapsed" type="xsd:boolean"
  use="optional" default="false">
  <xsd:annotation>
    <xsd:documentation>Collapsed</xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
</xsd:complexType>

<xsd:complexType name="CT_Cell">
  <xsd:attribute name="s" type="xsd:unsignedInt" use="optional"
    default="0">
    <xsd:annotation>
      <xsd:documentation>Style Index</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>

<!-- Cell Hidden on Lock in styles*.xml -->
<xsd:element name="styleSheet" type="CT_Stylesheet">
  <xsd:annotation>
    <xsd:documentation>Style Sheet</xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:complexType name="CT_Stylesheet">
  <xsd:sequence>
    <xsd:element name="cellXfs" type="CT_CellXfs"
      minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Cell Formats</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="cellStyles" type="CT_CellStyles"
      minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Cell Styles</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CT_CellXfs">
  <xsd:sequence>
    <xsd:element name="xf" type="CT_Xf" minOccurs="1"
      maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation>Format</xsd:documentation>
      </xsd:annotation>

```

```

        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Xf">
      <xsd:sequence>
        <xsd:element name="protection" type="CT_CellProtection"
          minOccurs="0" maxOccurs="1">
          <xsd:annotation>
            <xsd:documentation>Protection</xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="CT_CellProtection">
      <xsd:attribute name="locked" type="xsd:boolean" use="optional">
        <xsd:annotation>
          <xsd:documentation>Cell Locked</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="hidden" type="xsd:boolean" use="optional">
        <xsd:annotation>
          <xsd:documentation>Hidden Cell</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    </xsd:complexType>

    <xsd:complexType name="CT_CellStyles">
      <xsd:sequence>
        <xsd:element name="cellStyle" type="CT_CellStyle"
          minOccurs="1" maxOccurs="unbounded">
          <xsd:annotation>
            <xsd:documentation>Cell Style</xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
      <xsd:attribute name="count" type="xsd:unsignedInt"
        use="optional">
        <xsd:annotation>
          <xsd:documentation>Style Count</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    </xsd:complexType>

    <xsd:complexType name="CT_CellStyle">
      <xsd:attribute name="hidden" type="xsd:boolean" use="optional">
        <xsd:annotation>
          <xsd:documentation>Hidden Style</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    </xsd:complexType>

    <!-- hidden scenarios in sheet*.xml -->
    <xsd:complexType name="CT_Scenarios">
      <xsd:sequence>
        <xsd:element name="scenario" type="CT_Scenario"

```

```

        minOccurs="1" maxOccurs="unbounded">
        <xsd:annotation>
            <xsd:documentation>Scenario</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CT_Scenario">
    <xsd:attribute name="hidden" type="xsd:boolean" use="optional"
        default="false">
        <xsd:annotation>
            <xsd:documentation>Hidden Scenario</xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
</xsd:complexType>

<!-- Hidden, and veryHidden, sheets in workbook xml -->
<xsd:element name="workbook" type="CT_Workbook">
    <xsd:annotation>
        <xsd:documentation>Workbook</xsd:documentation>
    </xsd:annotation>
</xsd:element>

<xsd:complexType name="CT_Workbook">
    <xsd:sequence>
        <xsd:element name="sheets" type="CT_Sheets" minOccurs="1"
            maxOccurs="1">
            <xsd:annotation>
                <xsd:documentation>Sheets</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CT_Sheets">
    <xsd:sequence>
        <xsd:element name="sheet" type="CT_Sheet" minOccurs="1"
            maxOccurs="unbounded">
            <xsd:annotation>
                <xsd:documentation>Sheet Information</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CT_Sheet">
    <xsd:attribute name="name" type="ST_Xstring" use="required">
        <xsd:annotation>
            <xsd:documentation>Sheet Name</xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="state" type="ST_SheetState" use="optional"
        default="visible">
        <xsd:annotation>
            <xsd:documentation>Visible State</xsd:documentation>

```

```

        </xsd:annotation>
      </xsd:attribute>
    </xsd:complexType>

    <xsd:simpleType name="ST_SheetState">
      <xsd:annotation>
        <xsd:documentation>Sheet Visibility Types</xsd:documentation>
      </xsd:annotation>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="visible">
          <xsd:annotation>
            <xsd:documentation>Visible</xsd:documentation>
          </xsd:annotation>
        </xsd:enumeration>
        <xsd:enumeration value="hidden">
          <xsd:annotation>
            <xsd:documentation>Hidden</xsd:documentation>
          </xsd:annotation>
        </xsd:enumeration>
        <xsd:enumeration value="veryHidden">
          <xsd:annotation>
            <xsd:documentation>Very Hidden</xsd:documentation>
          </xsd:annotation>
        </xsd:enumeration>
      </xsd:restriction>
    </xsd:simpleType>

```

PRODUCT: POWERPOINT

```

<!-- slide*.xml example of skiping slides in a presentation -->
<xsd:element name="sld" type="CT_Slide">
  <xsd:annotation>
    <xsd:documentation>Presentation Slide</xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:complexType name="CT_Slide">
  <xsd:attribute name="show" type="xsd:boolean" use="optional"
    default="true">
    <xsd:annotation>
      <xsd:documentation>Show Slide in Slide Show</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>

<!-- slide*.xml example of hiding content in Black and White settings
-->
<xsd:element name="sp" type="CT_Shape">
  <xsd:annotation>
    <xsd:documentation>Shape</xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:complexType name="CT_Shape">
  <xsd:sequence>
    <xsd:element name="spPr" type="CT_ShapeProperties"

```

```

        minOccurs="1" maxOccurs="1" />
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CT_ShapeProperties">
    <xsd:attribute name="bwMode" type="ST_BlackWhiteMode"
        use="optional">
        <xsd:annotation>
            <xsd:documentation>Black and White Mode</xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
</xsd:complexType>

<xsd:simpleType name="ST_BlackWhiteMode">
    <xsd:annotation>
        <xsd:documentation>Black and White Mode</xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:token">
        <xsd:enumeration value="clr">
            <xsd:annotation>
                <xsd:documentation>Color</xsd:documentation>
            </xsd:annotation>
        </xsd:enumeration>
        <xsd:enumeration value="auto">
            <xsd:annotation>
                <xsd:documentation>Automatic</xsd:documentation>
            </xsd:annotation>
        </xsd:enumeration>
        <xsd:enumeration value="gray">
            <xsd:annotation>
                <xsd:documentation>Gray</xsd:documentation>
            </xsd:annotation>
        </xsd:enumeration>
        <xsd:enumeration value="ltGray">
            <xsd:annotation>
                <xsd:documentation>Light Gray</xsd:documentation>
            </xsd:annotation>
        </xsd:enumeration>
        <xsd:enumeration value="invGray">
            <xsd:annotation>
                <xsd:documentation>Inverse Gray</xsd:documentation>
            </xsd:annotation>
        </xsd:enumeration>
        <xsd:enumeration value="grayWhite">
            <xsd:annotation>
                <xsd:documentation>Gray and White</xsd:documentation>
            </xsd:annotation>
        </xsd:enumeration>
        <xsd:enumeration value="blackGray">
            <xsd:annotation>
                <xsd:documentation>Black and Gray</xsd:documentation>
            </xsd:annotation>
        </xsd:enumeration>
        <xsd:enumeration value="blackWhite">
            <xsd:annotation>
                <xsd:documentation>Black and White</xsd:documentation>
            </xsd:annotation>
        </xsd:enumeration>
    </xsd:restriction>
</xsd:simpleType>

```



```

        </xsd:annotation>
      </xsd:enumeration>
      <xsd:enumeration value="black">
        <xsd:annotation>
          <xsd:documentation>Black</xsd:documentation>
        </xsd:annotation>
      </xsd:enumeration>
      <xsd:enumeration value="white">
        <xsd:annotation>
          <xsd:documentation>White</xsd:documentation>
        </xsd:annotation>
      </xsd:enumeration>
      <xsd:enumeration value="hidden">
        <xsd:annotation>
          <xsd:documentation>Hidden</xsd:documentation>
        </xsd:annotation>
      </xsd:enumeration>
    </xsd:restriction>
  </xsd:simpleType>

```

OFFICE 2007.5.13: **END**

OFFICE 2007.5.14: **Authors**

```

<xs:element name="coreProperties" type="CT_CoreProperties" />

<xs:element name="any" type="SimpleLiteral" abstract="true" />

<xs:complexType name="CT_CoreProperties">
  <xs:all>
    <xs:element name="creator" substitutionGroup="any" />
    <xs:element name="lastModifiedBy" minOccurs="0"
      maxOccurs="1" type="xs:string" />
  </xs:all>
</xs:complexType>

```

OFFICE 2007.5.14: **END**

OFFICE 2007.5.15: **Printer Information**

```

<!-- Schema for printer settings in Excel -->
<Part>
  <ContentType>application/vnd.openxmlformats-
officedocument.spreadsheetml.printerSettings</ContentType>
  <RootNamespace/>
  <SourceRelationship>http://schemas.openxmlformats.org/officeDocument/
2006/relationships/printerSettings</SourceRelationship>
</Part>

<!-- Schema for printer settings in Word -->

```

```

<Part>
  <ContentType>application/vnd.openxmlformats-
officedocument.wordprocessingml.printerSettings</ContentType>
  <RootNamespace/>
  <SourceRelationship>http://schemas.openxmlformats.org/officeDocument/
2006/relationships/printerSettings</SourceRelationship>
</Part>

```

OFFICE 2007.5.15: **END**

OFFICE 2007.5.16: **Smart Tags** **PRODUCT: WORD**

```

<xsd:element name="p" type="CT_P" minOccurs="0" maxOccurs="unbounded">
  <xsd:annotation>
    <xsd:documentation>Paragraph</xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:complexType name="CT_P">
  <xsd:sequence>
    <xsd:element name="smartTag" type="CT_SmartTagRun">
      <xsd:annotation>
        <xsd:documentation>Inline-Level Smart Tag</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CT_SmartTagRun">
  <xsd:sequence>
    <xsd:element name="smartTagPr" type="CT_SmartTagPr"
      minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Smart Tag Properties</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="r" type="CT_R">
      <xsd:annotation>
        <xsd:documentation>Text Run</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CT_R">
  <xsd:sequence>
    <xsd:element name="t" type="CT_Text">
      <xsd:annotation>
        <xsd:documentation>Text</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

```

```

</xsd:complexType>

<xsd:complexType name="CT_SmartTagPr">
  <xsd:sequence>
    <xsd:element name="attr" type="CT_Attr" minOccurs="0"
      maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation>Smart Tag Property</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CT_Attr">
  <xsd:attribute name="uri" type="ST_String">
    <xsd:annotation>
      <xsd:documentation>Namespace</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="name" type="ST_String" use="required">
    <xsd:annotation>
      <xsd:documentation>Name</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="val" type="ST_String" use="required">
    <xsd:annotation>
      <xsd:documentation>Value</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>

```

PRODUCT: EXCEL

```

<xsd:element name="workbook" type="CT_Workbook">
  <xsd:annotation>
    <xsd:documentation>Workbook</xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:complexType name="CT_Workbook">
  <xsd:sequence>
    <xsd:element name="smartTagPr" type="CT_SmartTagPr"
      minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Smart Tag Properties</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="smartTagTypes" type="CT_SmartTagTypes"
      minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Smart Tag Types</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

```

```

<xsd:element name="smartTagPr" type="CT_SmartTagPr"
  minOccurs="0" maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>Smart Tag Properties</xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:element name="smartTagTypes" type="CT_SmartTagTypes"
  minOccurs="0" maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>Smart Tag Types</xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:complexType name="CT_SmartTagPr">
  <xsd:attribute name="embed" type="xsd:boolean" use="optional"
    default="false">
    <xsd:annotation>
      <xsd:documentation>Embed SmartTags</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="show" type="ST_SmartTagShow" use="optional"
    default="all">
    <xsd:annotation>
      <xsd:documentation>Show Smart Tags</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>

<xsd:simpleType name="ST_SmartTagShow">
  <xsd:annotation>
    <xsd:documentation>Smart Tag Display Types</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="all">
      <xsd:annotation>
        <xsd:documentation>All</xsd:documentation>
      </xsd:annotation>
    </xsd:enumeration>
    <xsd:enumeration value="none">
      <xsd:annotation>
        <xsd:documentation>None</xsd:documentation>
      </xsd:annotation>
    </xsd:enumeration>
    <xsd:enumeration value="noIndicator">
      <xsd:annotation>
        <xsd:documentation>No Smart Tag Indicator</xsd:documentation>
      </xsd:annotation>
    </xsd:enumeration>
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="CT_SmartTagTypes">
  <xsd:sequence>
    <xsd:element name="smartTagType" type="CT_SmartTagType"
      minOccurs="0" maxOccurs="unbounded">

```

```

        <xsd:annotation>
            <xsd:documentation>Smart Tag Type</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CT_SmartTagType">
    <xsd:attribute name="namespaceUri" type="ST_Xstring"
        use="optional">
        <xsd:annotation>
            <xsd:documentation>SmartTag Namespace URI</xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="name" type="ST_Xstring" use="optional">
        <xsd:annotation>
            <xsd:documentation>Name</xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="url" type="ST_Xstring" use="optional">
        <xsd:annotation>
            <xsd:documentation>Smart Tag URL</xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
</xsd:complexType>

```

PRODUCT: POWERPOINT

```

<xsd:element name="presentation" type="CT_Presentation">
    <xsd:annotation>
        <xsd:documentation>Presentation</xsd:documentation>
    </xsd:annotation>
</xsd:element>

<xsd:complexType name="CT_Presentation">
    <xsd:sequence>
        <xsd:element name="smartTags" type="CT_SmartTags"
            minOccurs="0" maxOccurs="1">
            <xsd:annotation>
                <xsd:documentation>Smart Tags</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CT_SmartTags">
    <xsd:attribute ref="r:id" use="required">
        <xsd:annotation>
            <xsd:documentation>Relationship Identifier</xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
</xsd:complexType>

```

OFFICE 2007.5.16: **END**

OFFICE 2007.5.17: **Imagery**
PRODUCT: WORD, EXCEL, AND POWERPOINT

Shape with image background:

```
<xsd:element name="pict" type="CT_Picture">
  <xsd:annotation>
    <xsd:documentation>VML Object</xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:complexType name="CT_Picture">
  <xsd:complexContent>
    <xsd:extension base="CT_PictureBase">
      <xsd:sequence maxOccurs="1">
        <xsd:element name="movie" type="CT_Rel" minOccurs="0">
          <xsd:annotation>
            <xsd:documentation>Embedded Video</xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="control" type="CT_Control" minOccurs="0">
          <xsd:annotation>
            <xsd:documentation>Floating Embedded
Control</xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:element name="shape" type="CT_Shape">
  <xsd:annotation>
    <xsd:documentation>Shape Definition</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="shapetype" type="CT_Shapetype">
  <xsd:annotation>
    <xsd:documentation>Shape Template</xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:complexType name="CT_Shape">
  <xsd:choice maxOccurs="unbounded">
    <xsd:group ref="EG_ShapeElements" />
    <xsd:element ref="o:ink" />
    <xsd:element ref="p:iscomment" />
  </xsd:choice>
  <xsd:attributeGroup ref="AG_AllCoreAttributes" />
  <xsd:attributeGroup ref="AG_AllShapeAttributes" />
  <xsd:attributeGroup ref="AG_Type" />

```

```

<xsd:attributeGroup ref="AG_Adj" />
<xsd:attributeGroup ref="AG_Path" />
<xsd:attribute ref="o:gfxdata">
  <xsd:annotation>
    <xsd:documentation>Encoded Package</xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
<xsd:attribute name="equationxml" type="xsd:string" use="optional">
  <xsd:annotation>
    <xsd:documentation>Storage for Alternate Math
Content</xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
</xsd:complexType>

<xsd:group name="EG_ShapeElements">
  <xsd:choice>
    <xsd:element ref="path" />
    <xsd:element ref="formulas" />
    <xsd:element ref="handles" />
    <xsd:element ref="fill" />
    <xsd:element ref="stroke" />
    <xsd:element ref="shadow" />
    <xsd:element ref="textbox" />
    <xsd:element ref="textpath" />
    <xsd:element ref="imagedata" />
    <xsd:element ref="o:skew" />
    <xsd:element ref="o:extrusion" />
    <xsd:element ref="o:callout" />
    <xsd:element ref="o:lock" />
    <xsd:element ref="o:clippath" />
    <xsd:element ref="o:signatureline" />
    <xsd:element ref="wvml:wrap" />
    <xsd:element ref="wvml:anchorlock" />
    <xsd:element ref="wvml:bordertop" />
    <xsd:element ref="wvml:borderbottom" />
    <xsd:element ref="wvml:borderleft" />
    <xsd:element ref="wvml:borderright" />
    <xsd:element ref="x:ClientData" minOccurs="0" />
    <xsd:element ref="p:textdata" minOccurs="0" />
  </xsd:choice>
</xsd:group>

<xsd:element name="fill" type="CT_Fill">
  <xsd:annotation>
    <xsd:documentation>Shape Fill Properties</xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:complexType name="CT_Fill">
  <xsd:sequence>
    <xsd:element ref="o:fill" minOccurs="0" />
  </xsd:sequence>
  <xsd:attributeGroup ref="AG_Id" />
  <xsd:attribute name="type" type="ST_FillType" use="optional">
    <xsd:annotation>

```



```

        <xsd:documentation>Fill Type</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="on" type="ST_TrueFalse" use="optional">
      <xsd:annotation>
        <xsd:documentation>Fill Toggle</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="color" type="ST_ColorType" use="optional">
      <xsd:annotation>
        <xsd:documentation>Primary Color</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="opacity" type="xsd:string" use="optional">
      <xsd:annotation>
        <xsd:documentation>Primary Color Opacity</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="color2" type="ST_ColorType" use="optional">
      <xsd:annotation>
        <xsd:documentation>Secondary Color</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="src" type="xsd:string" use="optional">
      <xsd:annotation>
        <xsd:documentation>Fill Image Source</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute ref="o:href">
      <xsd:annotation>
        <xsd:documentation>Hyperlink Target</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute ref="o:althref">
      <xsd:annotation>
        <xsd:documentation>Alternate Image Reference
Location</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="size" type="xsd:string" use="optional">
      <xsd:annotation>
        <xsd:documentation>Fill Image Size</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="origin" type="xsd:string" use="optional">
      <xsd:annotation>
        <xsd:documentation>Fill Image Origin</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="position" type="xsd:string" use="optional">
      <xsd:annotation>
        <xsd:documentation>Fill Image Position</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="aspect" type="ST_ImageAspect" use="optional">
      <xsd:annotation>

```

```

        <xsd:documentation>Image Aspect Ratio</xsd:documentation>
    </xsd:annotation>
</xsd:attribute>
<xsd:attribute name="colors" type="xsd:string" use="optional">
    <xsd:annotation>
        <xsd:documentation>Intermediate Colors</xsd:documentation>
    </xsd:annotation>
</xsd:attribute>
<xsd:attribute name="angle" type="xsd:decimal" use="optional">
    <xsd:annotation>
        <xsd:documentation>Gradient Angle</xsd:documentation>
    </xsd:annotation>
</xsd:attribute>
<xsd:attribute name="alignshape" type="ST_TrueFalse" use="optional">
    <xsd:annotation>
        <xsd:documentation>Align Image With Shape</xsd:documentation>
    </xsd:annotation>
</xsd:attribute>
<xsd:attribute name="focus" type="xsd:string" use="optional">
    <xsd:annotation>
        <xsd:documentation>Gradient Center</xsd:documentation>
    </xsd:annotation>
</xsd:attribute>
<xsd:attribute name="focussize" type="xsd:string" use="optional">
    <xsd:annotation>
        <xsd:documentation>Radial Gradient Size</xsd:documentation>
    </xsd:annotation>
</xsd:attribute>
<xsd:attribute name="focusposition" type="xsd:string"
use="optional">
    <xsd:annotation>
        <xsd:documentation>Radial Gradient Center</xsd:documentation>
    </xsd:annotation>
</xsd:attribute>
<xsd:attribute name="method" type="ST_FillMethod" use="optional">
    <xsd:annotation>
        <xsd:documentation>Gradient Fill Method</xsd:documentation>
    </xsd:annotation>
</xsd:attribute>
<xsd:attribute ref="o:detectmouseclick">
    <xsd:annotation>
        <xsd:documentation>Detect Mouse Click</xsd:documentation>
    </xsd:annotation>
</xsd:attribute>
<xsd:attribute ref="o:title">
    <xsd:annotation>
        <xsd:documentation>Title</xsd:documentation>
    </xsd:annotation>
</xsd:attribute>
<xsd:attribute ref="o:opacity2">
    <xsd:annotation>
        <xsd:documentation>Secondary Color Opacity</xsd:documentation>
    </xsd:annotation>
</xsd:attribute>
<xsd:attribute name="recolor" type="ST_TrueFalse" use="optional">
    <xsd:annotation>

```

```

        <xsd:documentation>Recolor Fill as Picture</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="rotate" type="ST_TrueFalse" use="optional">
      <xsd:annotation>
        <xsd:documentation>Rotate Fill with Shape</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute ref="r:id" use="optional">
      <xsd:annotation>
        <xsd:documentation>Relationship to Part</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute ref="o:relid" use="optional">
      <xsd:annotation>
        <xsd:documentation>Relationship to Part</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  </xsd:complexType>

```

Chart with image background, SmartArt with image background:

```

<xsd:element name="blipFill" type="CT_BlipFillProperties"
minOccurs="1" maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>Picture Fill</xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:complexType name="CT_BlipFillProperties">
  <xsd:sequence>
    <xsd:element name="blip" type="CT_Blip" minOccurs="0"
maxOccurs="1"></xsd:element>
    <xsd:element name="srcRect" type="CT_RelativeRect" minOccurs="0"
maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Source Rectangle</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:group ref="EG_FillModeProperties" minOccurs="0" maxOccurs="1"
/>
  </xsd:sequence>
  <xsd:attribute name="dpi" type="xsd:unsignedInt" use="optional">
    <xsd:annotation>
      <xsd:documentation>DPI Setting</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="rotWithShape" type="xsd:boolean"
use="optional">
    <xsd:annotation>
      <xsd:documentation>Rotate With Shape</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>

```

```

<xsd:complexType name="CT_Blip">
  <xsd:sequence>
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
      <xsd:element name="alphaBiLevel" type="CT_AlphaBiLevelEffect"
minOccurs="1" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>Alpha Bi-Level Effect</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="alphaCeiling" type="CT_AlphaCeilingEffect"
minOccurs="1" maxOccurs="1" />
      <xsd:element name="alphaFloor" type="CT_AlphaFloorEffect"
minOccurs="1" maxOccurs="1" />
      <xsd:element name="alphaInv" type="CT_AlphaInverseEffect"
minOccurs="1" maxOccurs="1" />
      <xsd:element name="alphaMod" type="CT_AlphaModulateEffect"
minOccurs="1" maxOccurs="1" />
      <xsd:element name="alphaModFix"
type="CT_AlphaModulateFixedEffect" minOccurs="1" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>Alpha Modulate Fixed
Effect</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="alphaRepl" type="CT_AlphaReplaceEffect"
minOccurs="1" maxOccurs="1" />
      <xsd:element name="biLevel" type="CT_BiLevelEffect"
minOccurs="1" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>Bi-Level (Black/White)
Effect</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="blur" type="CT_BlurEffect" minOccurs="1"
maxOccurs="1" />
      <xsd:element name="clrChange" type="CT_ColorChangeEffect"
minOccurs="1" maxOccurs="1" />
      <xsd:element name="clrRepl" type="CT_ColorReplaceEffect"
minOccurs="1" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>Solid Color
Replacement</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="duotone" type="CT_DuotoneEffect"
minOccurs="1" maxOccurs="1" />
      <xsd:element name="fillOverlay" type="CT_FillOverlayEffect"
minOccurs="1" maxOccurs="1" />
      <xsd:element name="grayscale" type="CT_GrayscaleEffect"
minOccurs="1" maxOccurs="1" />
      <xsd:element name="hsl" type="CT_HSLEffect" minOccurs="1"
maxOccurs="1" />
      <xsd:element name="lum" type="CT_LuminanceEffect" minOccurs="1"
maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>Luminance Effect</xsd:documentation>

```

```

        </xsd:annotation>
      </xsd:element>
      <xsd:element name="tint" type="CT_TintEffect" minOccurs="1"
maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>Tint Effect</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:choice>
    <xsd:element name="extLst" type="CT_OfficeArtExtensionList"
minOccurs="0" maxOccurs="1"></xsd:element>
  </xsd:sequence>
  <xsd:attributeGroup ref="AG_Blob"></xsd:attributeGroup>
  <xsd:attribute name="cstate" type="ST_BlipCompression"
use="optional" default="none">
    <xsd:annotation>
      <xsd:documentation>Compression State</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>

<xsd:attributeGroup name="AG_Blob">
  <xsd:attribute ref="r:embed" use="optional" default="">
    <xsd:annotation>
      <xsd:documentation>Embedded Picture
Reference</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute ref="r:link" use="optional" default="">
    <xsd:annotation>
      <xsd:documentation>Linked Picture Reference</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:attributeGroup>

```

PRODUCT: WORD

Insert Picture or Clip Art:

See "Chart with image background" and "SmartArt with image background." above for **blipFill** and **blip** grammar.

Document background:

```

<xsd:element name="background" type="CT_Background" minOccurs="0">
  <xsd:annotation>
    <xsd:documentation>Document Background</xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:complexType name="CT_Background">
  <xsd:complexContent>
    <xsd:extension base="CT_PictureBase">
      <xsd:attribute name="color" type="ST_HexColor" use="optional">
        <xsd:annotation>
          <xsd:documentation>Background Color</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

```

        </xsd:annotation>
        </xsd:attribute>
        <xsd:attribute name="themeColor" type="ST_ThemeColor"
use="optional">
        <xsd:annotation>
        <xsd:documentation>Background Theme
Color</xsd:documentation>
        </xsd:annotation>
        </xsd:attribute>
        <xsd:attribute name="themeTint" type="ST_UcharHexNumber"
use="optional">
        <xsd:annotation>
        <xsd:documentation>Border Theme Color
Tint</xsd:documentation>
        </xsd:annotation>
        </xsd:attribute>
        <xsd:attribute name="themeShade" type="ST_UcharHexNumber"
use="optional">
        <xsd:annotation>
        <xsd:documentation>Border Theme Color
Shade</xsd:documentation>
        </xsd:annotation>
        </xsd:attribute>
        </xsd:extension>
        </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="CT_PictureBase">
    <xsd:sequence maxOccurs="unbounded">
        <xsd:any processContents="lax" namespace="urn:schemas-microsoft-
om:vml" minOccurs="0" />
        <xsd:any processContents="lax" namespace="urn:schemas-microsoft-
com:office:office" minOccurs="0" />
    </xsd:sequence>
</xsd:complexType>

<xsd:element name="background" type="CT_Background">
    <xsd:annotation>
        <xsd:documentation>Document Background</xsd:documentation>
    </xsd:annotation>
</xsd:element>

<xsd:complexType name="CT_Background">
    <xsd:sequence>
        <xsd:element ref="fill" minOccurs="0" />
    </xsd:sequence>
    <xsd:attributeGroup ref="AG_Id" />
    <xsd:attributeGroup ref="AG_Fill" />
    <xsd:attribute ref="o:bwmode">
        <xsd:annotation>
            <xsd:documentation>Black-and-White Mode</xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute ref="o:bwpure">
        <xsd:annotation>
            <xsd:documentation>Pure Black-and-White Mode</xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>

```

```

        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute ref="o:bwnormal">
        <xsd:annotation>
          <xsd:documentation>Normal Black-and-White
Mode</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute ref="o:targetscreensize">
        <xsd:annotation>
          <xsd:documentation>Target Screen Size</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    </xsd:complexType>

```

See Grammar on "Shape with image background, WordArt with image overlay" for **fill** and **CT_Fill** grammars.

Quick Parts - IncludePicture Field and Watermarks:

See "Shape with image background, WordArt with image overlay" above for grammars on the w:pict, v:shapetype and v:shape elements.

```

<xsd:element name="imagedata" type="CT_ImageData">
  <xsd:annotation>
    <xsd:documentation>Image Data</xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:complexType name="CT_ImageData">
  <xsd:attributeGroup ref="AG_Id" />
  <xsd:attributeGroup ref="AG_ImageAttributes" />
  <xsd:attributeGroup ref="AG_Chromakey" />
  <xsd:attribute name="embosscolor" type="ST_ColorType"
use="optional">
    <xsd:annotation>
      <xsd:documentation>Embossed Color</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="recolortarget" type="ST_ColorType">
    <xsd:annotation>
      <xsd:documentation>Black Recoloring Color</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute ref="o:href">
    <xsd:annotation>
      <xsd:documentation>Original Image Reference</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute ref="o:althref">
    <xsd:annotation>
      <xsd:documentation>Alternate Image Reference</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute ref="o:title">

```

```

        <xsd:annotation>
          <xsd:documentation>Image Data Title</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute ref="o:oleid">
        <xsd:annotation>
          <xsd:documentation>Image Embedded Object ID</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute ref="o:detectmouseclick">
        <xsd:annotation>
          <xsd:documentation>Detect Mouse Click</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute ref="o:movie">
        <xsd:annotation>
          <xsd:documentation>Movie Reference</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute ref="o:relid">
        <xsd:annotation>
          <xsd:documentation>Relationship to Part</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute ref="r:id">
        <xsd:annotation>
          <xsd:documentation>Explicit Relationship to Image
Data</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute ref="r:pict">
        <xsd:annotation>
          <xsd:documentation>Explicit Relationship to Alternate Image
Data</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute ref="r:href">
        <xsd:annotation>
          <xsd:documentation>Explicit Relationship to Hyperlink
Target</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    </xsd:complexType>

    <xsd:attributeGroup name="AG_Id">
      <xsd:attribute name="id" type="xsd:string" use="optional">
        <xsd:annotation>
          <xsd:documentation>Unique Identifier</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    </xsd:attributeGroup>

```

WordArt with image overlay:

See grammar for "Shape with image background" above.

PRODUCT: EXCEL

Insert Picture or Clip Art:

```
<xsd:element name="drawing" type="CT_Drawing" minOccurs="0"
maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>Drawing</xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:complexType name="CT_Drawing">
  <xsd:attribute ref="r:id" use="required">
    <xsd:annotation>
      <xsd:documentation>Relationship id</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>
```

Document Background:

```
<xsd:element name="picture" type="CT_SheetBackgroundPicture"
minOccurs="0" maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>Background Image</xsd:documentation>
  </xsd:annotation>
</xsd:element>
```

WordArt with image overlay or image background:

See grammar for "Chart with image background, SmartArt with image background" above. for grammar details on blipFill and blip elements.

PRODUCT: POWERPOINT

Insert Picture or Clip Art , Docment Background, WordArt with image overlay or image background:

See grammar for "Chart with image background, SmartArt with image background" above. for grammar details on blipFill and blip elements.

OFFICE 2007.5.17: **END**

OFFICE 2007.5.18: **Database Connections and Queries**

PRODUCT: WORD

(abridged schema)

```
<xsd:complexType name="CT_MailMerge">
  <xsd:sequence>
    <xsd:element name="linkToQuery" type="CT_OnOff" minOccurs="0">
      <xsd:annotation>
```

```

        <xsd:documentation>Query Contains Link to External Query
File</xsd:documentation>
    </xsd:annotation>
</xsd:element>
    <xsd:element name="connectString" type="CT_String" minOccurs="0">
        <xsd:annotation>
            <xsd:documentation>Data Source Connection
String</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="query" type="CT_String" minOccurs="0">
        <xsd:annotation>
            <xsd:documentation>Query For Data Source Records To
Merge</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="dataSource" type="CT_Rel" minOccurs="0">
        <xsd:annotation>
            <xsd:documentation>Data Source File Path</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="headerSource" type="CT_Rel" minOccurs="0">
        <xsd:annotation>
            <xsd:documentation>Header Definition File
Path</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="odso" type="CT_Odso" minOccurs="0">
        <xsd:annotation>
            <xsd:documentation>Office Data Source Object
Settings</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CT_Odso">
    <xsd:sequence>
        <xsd:element name="udl" type="CT_String" minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>UDL Connection String</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="table" type="CT_String" minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Data Source Table Name</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="src" type="CT_Rel" minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>ODSO Data Source File
Path</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="fieldMapData" type="CT_OdsoFieldMapData"
minOccurs="0" maxOccurs="unbounded">

```

```

        <xsd:annotation>
          <xsd:documentation>External Data Source to Merge Field
Mapping</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="recipientData" type="CT_Rel" minOccurs="0"
maxOccurs="unbounded">
        <xsd:annotation>
          <xsd:documentation>Reference to Inclusion/Exclusion Data for
Data Source</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="CT_OdsoFieldMapData">
    <xsd:sequence>
      <xsd:element name="name" type="CT_String" minOccurs="0">
        <xsd:annotation>
          <xsd:documentation>Data Source Name for
Column</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>

```

PRODUCT: EXCEL

Complete grammar for connections can be found in “sml-externalConnections.xsd” within Office Open XML 1st edition Part 4’s XML Schemas.

Complete grammar for query tables can be found in “sml-queryTable.xsd” within Office Open XML 1st edition Part 4’s XML Schemas.

OFFICE 2007.5.18: **END**

OFFICE 2007.5.19: Tracked Changes**PRODUCT: WORD**

```

      <xsd:element name="moveFromRangeStart" type="CT_MoveBookmark">
        <xsd:annotation>
          <xsd:documentation>Move Source Location Container - Start
        </xsd:documentation>
        </xsd:annotation>
      </xsd:element>

      <xsd:element name="moveFromRangeEnd" type="CT_MarkupRange">
        <xsd:annotation>
          <xsd:documentation>Move Source Location Container - End
        </xsd:documentation>
        </xsd:annotation>
      </xsd:element>

```

```

<xsd:element name="moveToRangeStart" type="CT_MoveBookmark">
  <xsd:annotation>
    <xsd:documentation>Move Destination Location Container - Start
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:element name="moveToRangeEnd" type="CT_MarkupRange">
  <xsd:annotation>
    <xsd:documentation>Move Destination Location Container - End
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:complexType name="CT_MoveBookmark">
  <xsd:complexContent>
    <xsd:extension base="CT_Bookmark">
      <xsd:attribute name="author" type="ST_String" use="required">
        <xsd:annotation>
          <xsd:documentation>Annotation Author</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="date" type="ST_DateTime" use="required">
        <xsd:annotation>
          <xsd:documentation>Annotation Date</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="CT_Bookmark">
  <xsd:complexContent>
    <xsd:extension base="CT_BookmarkRange">
      <xsd:attribute name="name" type="ST_String" use="required">
        <xsd:annotation>
          <xsd:documentation>Bookmark Name</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="CT_BookmarkRange">
  <xsd:complexContent>
    <xsd:extension base="CT_MarkupRange">
      <xsd:attribute name="colFirst" type="ST_DecimalNumber"
        use="optional">
        <xsd:annotation>
          <xsd:documentation>First Table Column Covered By Bookmark
          </xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="colLast" type="ST_DecimalNumber"
        use="optional">

```

```

        <xsd:annotation>
            <xsd:documentation>Last Table Column Covered By Bookmark
        </xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="CT_MarkupRange">
    <xsd:complexContent>
        <xsd:extension base="CT_Markup">
            <xsd:attribute name="displacedByCustomXml"
type="ST_DisplacedByCustomXml"
use="optional">
                <xsd:annotation>
                    <xsd:documentation>Annotation Marker Relocated For Custom
                        XML Markup</xsd:documentation>
                </xsd:annotation>
            </xsd:attribute>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:group name="EG_ParaRPrTrackChanges">
    <xsd:sequence>
        <xsd:element name="ins" type="CT_TrackChange"
minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Inserted Paragraph</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="del" type="CT_TrackChange"
minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Deleted Paragraph</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="moveFrom" type="CT_TrackChange"
minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Move Source Paragraph</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="moveTo" type="CT_TrackChange"
minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Move Destination Paragraph
            </xsd:documentation>
            </xsd:annotation>
        </xsd:element>
    </xsd:sequence>
</xsd:group>

<xsd:complexType name="CT_TrackChange">
    <xsd:complexContent>

```

```

<xsd:extension base="CT_Markup">
  <xsd:attribute name="author" type="ST_String" use="required">
    <xsd:annotation>
      <xsd:documentation>Annotation Author</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="date" type="ST_DateTime" use="optional">
    <xsd:annotation>
      <xsd:documentation>Annotation Date</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="CT_Markup">
  <xsd:attribute name="id" type="ST_DecimalNumber" use="required">
    <xsd:annotation>
      <xsd:documentation>Annotation Identifier</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>

<!-- Tracked Style changes in document xml -->
<xsd:element name="rPr" type="CT_RPr">
  <xsd:annotation>
    <xsd:documentation>Run Properties</xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:complexType name="CT_RPr">
  <xsd:element name="rPrChange" type="CT_RPrChange"
    minOccurs="0">
    <xsd:annotation>
      <xsd:documentation>Revision Information for Run Properties
    </xsd:documentation>
    </xsd:annotation>
  </xsd:element>
</xsd:complexType>

<xsd:complexType name="CT_RPrChange">
  <xsd:complexContent>
    <xsd:extension base="CT_TrackChange">
      <xsd:sequence>
        <xsd:element name="rPr" type="CT_RPr"
          minOccurs="1">
          <xsd:annotation>
            <xsd:documentation>Previous Run Properties
          </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

PRODUCT: EXCEL

See MS sml-sharedWorkbookRevisions schema for more information.

OFFICE 2007.5.19: **END**

OFFICE 2007.5.20: **Document Security**
PRODUCT: WORD, EXCEL, AND POWERPOINT

Encryption:

See [MS-OFFCRYPTO].pdf, Section 1.3.1, for an illustration of the Data Spaces structure used when storing content in OLE compound files.

Digital Signatures:

Complete schema for xml signatures can be found at
["http://www.w3.org/2000/09/xmldsig"](http://www.w3.org/2000/09/xmldsig).

PRODUCT: WORD

Write Reservation Password:

For w:writeProtection:

```
<xsd:complexType name="CT_Settings">
  <xsd:sequence>
    <xsd:element name="writeProtection" type="CT_WriteProtection"
minOccurs="0">
      <xsd:annotation>
        <xsd:documentation>Write Protection</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CT_WriteProtection">
  <xsd:attribute name="recommended" type="ST_OnOff" use="optional">
    <xsd:annotation>
      <xsd:documentation>Recommend Write Protection in User
Interface</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attributeGroup ref="AG_Password"></xsd:attributeGroup>
</xsd:complexType>

<xsd:attributeGroup name="AG_Password">
  <xsd:attribute name="cryptProviderType" type="ST_CryptProv">
    <xsd:annotation>
```

```

        <xsd:documentation>Cryptographic Provider
Type</xsd:documentation>
    </xsd:annotation>
</xsd:attribute>
    <xsd:attribute name="cryptAlgorithmClass" type="ST_AlgClass">
    <xsd:annotation>
        <xsd:documentation>Cryptographic Algorithm
Class</xsd:documentation>
    </xsd:annotation>
</xsd:attribute>
    <xsd:attribute name="cryptAlgorithmType" type="ST_AlgType">
    <xsd:annotation>
        <xsd:documentation>Cryptographic Algorithm
Type</xsd:documentation>
    </xsd:annotation>
</xsd:attribute>
    <xsd:attribute name="cryptAlgorithmSid" type="ST_DecimalNumber">
    <xsd:annotation>
        <xsd:documentation>Cryptographic Hashing
Algorithm</xsd:documentation>
    </xsd:annotation>
</xsd:attribute>
    <xsd:attribute name="cryptSpinCount" type="ST_DecimalNumber">
    <xsd:annotation>
        <xsd:documentation>Iterations to Run Hashing
Algorithm</xsd:documentation>
    </xsd:annotation>
</xsd:attribute>
    <xsd:attribute name="cryptProvider" type="ST_String">
    <xsd:annotation>
        <xsd:documentation>Cryptographic Provider</xsd:documentation>
    </xsd:annotation>
</xsd:attribute>
    <xsd:attribute name="algIdExt" type="ST_LongHexNumber">
    <xsd:annotation>
        <xsd:documentation>Cryptographic Algorithm
Extensibility</xsd:documentation>
    </xsd:annotation>
</xsd:attribute>
    <xsd:attribute name="algIdExtSource" type="ST_String">
    <xsd:annotation>
        <xsd:documentation>Algorithm Extensibility
Source</xsd:documentation>
    </xsd:annotation>
</xsd:attribute>
    <xsd:attribute name="cryptProviderTypeExt" type="ST_LongHexNumber">
    <xsd:annotation>
        <xsd:documentation>Cryptographic Provider Type
Extensibility</xsd:documentation>
    </xsd:annotation>
</xsd:attribute>
    <xsd:attribute name="cryptProviderTypeExtSource" type="ST_String">
    <xsd:annotation>
        <xsd:documentation>Provider Type Extensibility
Source</xsd:documentation>
    </xsd:annotation>

```



```

</xsd:attribute>
<xsd:attribute name="hash" type="xsd:base64Binary">
  <xsd:annotation>
    <xsd:documentation>Password Hash</xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
<xsd:attribute name="salt" type="xsd:base64Binary">
  <xsd:annotation>
    <xsd:documentation>Salt for Password
Verifier</xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
</xsd:attributeGroup>

```

For w:documentProtection:

```

<xsd:element name="documentProtection" type="CT_DocProtect"
minOccurs="0">
  <xsd:annotation>
    <xsd:documentation>Document Editing
Restrictions</xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:complexType name="CT_DocProtect">
  <xsd:attribute name="edit" type="ST_DocProtect" use="optional">
    <xsd:annotation>
      <xsd:documentation>Document Editing
Restrictions</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="formatting" type="ST_OnOff" use="optional">
    <xsd:annotation>
      <xsd:documentation>Only Allow Formatting With Unlocked
Styles</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="enforcement" type="ST_OnOff">
    <xsd:annotation>
      <xsd:documentation>Enforce Document Protection
Settings</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attributeGroup ref="AG_Password"></xsd:attributeGroup>
</xsd:complexType>

<xsd:element name="permStart" minOccurs="0" type="CT_PermStart">
  <xsd:annotation>
    <xsd:documentation>Range Permission Start</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="permEnd" minOccurs="0" type="CT_Perm">
  <xsd:annotation>
    <xsd:documentation>Range Permission End</xsd:documentation>
  </xsd:annotation>

```

```

</xsd:annotation>
</xsd:element>
<xsd:complexType name="CT_Perm">
  <xsd:attribute name="id" type="ST_String" use="required">
    <xsd:annotation>
      <xsd:documentation>Annotation ID</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="displacedByCustomXml"
type="ST_DisplacedByCustomXml" use="optional">
    <xsd:annotation>
      <xsd:documentation>Annotation Displaced By Custom XML
Markup</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>
<xsd:complexType name="CT_PermStart">
  <xsd:complexContent>
    <xsd:extension base="CT_Perm">
      <xsd:attribute name="edGrp" type="ST_EdGrp" use="optional">
        <xsd:annotation>
          <xsd:documentation>Editor Group For Range
Permission</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="ed" type="ST_String" use="optional">
        <xsd:annotation>
          <xsd:documentation>Single User For Range
Permission</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="colFirst" type="ST_DecimalNumber"
use="optional">
        <xsd:annotation>
          <xsd:documentation>First Table Column Covered By Range
Permission</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="colLast" type="ST_DecimalNumber"
use="optional">
        <xsd:annotation>
          <xsd:documentation>Last Table Column Covered By Range
Permission</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

PRODUCT: EXCEL

Write Reservation Password:

fileSharing:

```

<xsd:element name="fileSharing" type="CT_FileSharing" minOccurs="0"
maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>File Sharing</xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:complexType name="CT_FileSharing">
  <xsd:attribute name="readOnlyRecommended" type="xsd:boolean"
use="optional" default="false">
    <xsd:annotation>
      <xsd:documentation>Read Only Recommended</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="userName" type="ST_Xstring">
    <xsd:annotation>
      <xsd:documentation>User Name</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="reservationPassword"
type="ST_UnsignedShortHex">
    <xsd:annotation>
      <xsd:documentation>Write Reservation
Password</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>

```

sheetProtection:

```

<xsd:element name="sheetProtection" type="CT_SheetProtection"
minOccurs="0" maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>Sheet Protection Options</xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:complexType name="CT_SheetProtection">
  <xsd:attribute name="password" type="ST_UnsignedShortHex"
use="optional">
    <xsd:annotation>
      <xsd:documentation>Password</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="sheet" type="xsd:boolean" use="optional"
default="false">
    <xsd:annotation>
      <xsd:documentation>Sheet Locked</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="objects" type="xsd:boolean" use="optional"
default="false">
    <xsd:annotation>
      <xsd:documentation>Objects Locked</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>

```

```

        </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="scenarios" type="xsd:boolean" use="optional"
default="false">
        <xsd:annotation>
            <xsd:documentation>Scenarios Locked</xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="formatCells" type="xsd:boolean" use="optional"
default="true">
        <xsd:annotation>
            <xsd:documentation>Format Cells Locked</xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="formatColumns" type="xsd:boolean"
use="optional" default="true">
        <xsd:annotation>
            <xsd:documentation>Format Columns Locked</xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="formatRows" type="xsd:boolean" use="optional"
default="true">
        <xsd:annotation>
            <xsd:documentation>Format Rows Locked</xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="insertColumns" type="xsd:boolean"
use="optional" default="true">
        <xsd:annotation>
            <xsd:documentation>Insert Columns Locked</xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="insertRows" type="xsd:boolean" use="optional"
default="true">
        <xsd:annotation>
            <xsd:documentation>Insert Rows Locked</xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="insertHyperlinks" type="xsd:boolean"
use="optional" default="true">
        <xsd:annotation>
            <xsd:documentation>Insert Hyperlinks Locked</xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="deleteColumns" type="xsd:boolean"
use="optional" default="true">
        <xsd:annotation>
            <xsd:documentation>Delete Columns Locked</xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="deleteRows" type="xsd:boolean" use="optional"
default="true">
        <xsd:annotation>
            <xsd:documentation>Delete Rows Locked</xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>

```

```

    <xsd:attribute name="selectLockedCells" type="xsd:boolean"
use="optional" default="false">
    <xsd:annotation>
    <xsd:documentation>Select Locked Cells
Locked</xsd:documentation>
    </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="sort" type="xsd:boolean" use="optional"
default="true">
    <xsd:annotation>
    <xsd:documentation>Sort Locked</xsd:documentation>
    </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="autoFilter" type="xsd:boolean" use="optional"
default="true">
    <xsd:annotation>
    <xsd:documentation>AutoFilter Locked</xsd:documentation>
    </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="pivotTables" type="xsd:boolean" use="optional"
default="true">
    <xsd:annotation>
    <xsd:documentation>Pivot Tables Locked</xsd:documentation>
    </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="selectUnlockedCells" type="xsd:boolean"
use="optional" default="false">
    <xsd:annotation>
    <xsd:documentation>Select Unlocked Cells
Locked</xsd:documentation>
    </xsd:annotation>
    </xsd:attribute>
</xsd:complexType>

```

PRODUCT: POWERPOINT

Write Reservation Password:

```

<xsd:element name="modifyVerifier" type="CT_ModifyVerifier"
minOccurs="0" maxOccurs="1">
    <xsd:annotation>
    <xsd:documentation>Modification Verifier</xsd:documentation>
    </xsd:annotation>
</xsd:element>

<xsd:complexType name="CT_ModifyVerifier">
    <xsd:attribute name="cryptProviderType" type="ST_CryptProv"
use="required">
    <xsd:annotation>
    <xsd:documentation>Cryptographic Provider
Type</xsd:documentation>
    </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="cryptAlgorithmClass" type="ST_AlgClass"
use="required">
    <xsd:annotation>

```

```

        <xsd:documentation>Cryptographic Algorithm
Class</xsd:documentation>
    </xsd:annotation>
</xsd:attribute>
    <xsd:attribute name="cryptAlgorithmType" type="ST_AlgType"
use="required">
        <xsd:annotation>
            <xsd:documentation>Cryptographic Algorithm
Type</xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="cryptAlgorithmSid" type="xsd:unsignedInt "
use="required">
        <xsd:annotation>
            <xsd:documentation>Cryptographic Hashing
Algorithm</xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="spinCount" type="xsd:unsignedInt "
use="required">
        <xsd:annotation>
            <xsd:documentation>Iterations to Run Hashing
Algorithm</xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="saltData" type="xsd:string" use="required">
        <xsd:annotation>
            <xsd:documentation>Salt for Password
Verifier</xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="hashData" type="xsd:string" use="required">
        <xsd:annotation>
            <xsd:documentation>Password Hash</xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="cryptProvider" type="xsd:string"
use="optional">
        <xsd:annotation>
            <xsd:documentation>Cryptographic Provider</xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="algIdExt" type="xsd:unsignedInt "
use="optional">
        <xsd:annotation>
            <xsd:documentation>Cryptographic Algorithm
Extensibility</xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="algIdExtSource" type="xsd:string"
use="optional">
        <xsd:annotation>
            <xsd:documentation>Algorithm Extensibility
Source</xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>

```

```
<xsd:attribute name="cryptProviderTypeExt" type="xsd:unsignedInt"
use="optional">
  <xsd:annotation>
    <xsd:documentation>Cryptographic Provider Type
Extensibility</xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
<xsd:attribute name="cryptProviderTypeExtSource" type="xsd:string"
use="optional">
  <xsd:annotation>
    <xsd:documentation>Provider Type Extensibility
Source</xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
</xsd:complexType>
```

OFFICE 2007.5.20: **END**

6. ACRONYMS

The following table lists the acronyms that appear in this document.

Table 6-1 Acronyms

Acronym	Denotation
aBNF	augmented Backus-Naur Form
AES	Advanced Encryption Standard
API	Application Programming Interface
AR	All Reference
BLIP	Binary Large Image or Picture
BMP	Bitmap
BNF	Backus-Naur Form
BOF	Beginning of File
CDF	Compound Document Format
CDS	Cross Domain Solutions
CFB	Compound File Binary
CRC	Cyclic Redundancy Check
DIB	Device-Independent Bitmap
DTG	Data Transfer Guidance
EMF	Enhanced Metafile
EOF	End of File
ER	Excel Reference
FIB	File Information Block
GIF	Graphics Interchange Format
GUID	Globally Unique Identifier
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IEEE	Institute of Electrical and Electronics
ISG	Inspection and Sanitization Guidance
JPEG	Joint Photographic Experts Group Image
LTR	Left-to-Right
MOICE	Microsoft Office Isolate Conversion Environment

Acronym	Denotation
MS	Microsoft
NTFS	New Technology File System
OASIS	Organization for the Advancement of Structured Information Standards
OCR	Optical Character Recognition
ODF	Open Document Format
ODBC	Open Database Connectivity
ODSO	Office Data Source Object
OLAP	Online Analytical Processing
OLE	Object Linking and Embedding
OOXML	Office Open eXtensible Markup Language
OPC	Open Packaging Convention
PDF	Portable Document Format
PNG	Portable Network Graphics
POI	Poor Obfuscated Implementation
PR	PowerPoint Reference
RTL	Right-to-Left
SHA	Secure Hash Algorithm
SNAC	Systems and Network Analysis Center
SP	Service Pack
SQL	Structured Query Language
SVG	Scalable Vector Graphics
TIFF	Tagged Information File Format
UDL	Universal Data Link
URL	Uniform Resource Locator
UTF	Unicode Transformation Format
WMF	Windows Metafile
WR	Word Reference
XML	eXtensible Markup Language
XSD	XML Schema Definition

7. REFERENCED DOCUMENTS

The following documents were referenced in this document.

[1] Google Blog

<https://googleblog.blogspot.com/2007/04/were-expecting.html>

[2] Binary File Format Availability

<https://support.microsoft.com/kb/840817/en-us#>

[3] Microsoft's Binary File Format Specifications

<https://msdn.microsoft.com/en-us/library/cc313105.aspx>

[4] Oracle Clean Content

<https://www.oracle.com/technology/products/content-management/oit/cleancontent.html>

[5] Report # C4-072R-99: Microsoft Office 97 Executable Content Security Risks and Countermeasures

https://www.nsa.gov/snac/app/archived/off97_ec_security.pdf

[6] Report # C4-07R-02: Microsoft Office 2000 Executable Content Security Risks and Countermeasures

<https://www.nsa.gov/snac/app/C4-07R-02.PDF>

[7] Report # I333-001-05: Microsoft Office XP/2003 Executable Content Security Risks and Countermeasures

<https://www.nsa.gov/snac/app/I33-001-05.PDF>

[8] MOICE Advisory

<https://www.microsoft.com/technet/security/advisory/937696.msp>

[9] SNAC Fact Sheet for MOICE

<https://www.nsa.gov/snac/factsheets/I733-006R-2008.pdf>

[10] Office File Extensions

<https://office.microsoft.com/en-us/help/HA100069351033.aspx>

Appendix A

BNF Grammar

APPENDIX A: BNF GRAMMAR

Backus-Naur Form Grammar

Explanation of Construct Grammar

This document uses the “augmented Backus-Naur Form” (aBNF) to describe the grammar of constructs in the zip file. Because the document describes binary data, it extends the grammar definitions of standard aBNF. This extension should provide the reader with a sufficiently accurate portrayal of the binary data. In particular, the user should note the following:

- Comments begin with ‘<!--’ and close with ‘-->’, rather than proceeding them 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’).

Many Office 2007 constructs use similar data structures, especially for low level data such as strings and integers. This section defines some of these common structures.

```

<Integer64> ::= <octet>[8]

<Integer32> ::= <octet>[4]

<Integer16> ::= <octet>[2]

<Integer8> ::= <octet>

<octet> ::= <bit>[8]

<bit> ::= 0 | 1

```

OFFICE 2007.A.1: **Local File Header 1****PRODUCT: WORD, EXCEL, AND POWERPOINT****GRAMMAR:**

```

<Local File Header> ::= <Local File Header Signature> <Version Needed to
    Extract> <General Purpose Bit Flag> <Compression Method> <Last Mod
    File Time> <Last Mod File Date> <CRC-32><Compressed Size>
    <Uncompressed Size><File Name Length> <Extra Field Length> <File
    Name> <Extra Field>

```

```

<Local File Header Signature> ::= 0x04034b50

```

```

<Version Needed to Extract> ::= <Integer16>

```

```

<General Purpose Bit Flag> ::= <Integer16>

```

```

<Compression Method> ::= <Integer16>

```

```

<Last Mod File Time> ::= <Integer16>

```

```

<Last Mod File Date> ::= <Integer16>

```

```

<CRC-32> ::= <Integer32>

```

```

<Compressed Size> ::= <Integer32>

```

```

<Uncompressed Size> ::= <Integer32>

```

```

<File Name Length> ::= <Integer16>

```

```

<Extra Field Length> ::= <Integer16>

```

```

<File Name> ::= <octet>[File Name Length]

```

```

<Extra Field> ::= <octet>[Extra Field Length]

```

OFFICE 2007.A.1: **END****File Data 1: N/A**OFFICE 2007.A.2: **END****Central Directory Structure**

DESCRIPTION:
PRODUCT: WORD, EXCEL, POWERPOINT

GRAMMAR:

File Header:

```
<Central File Header> ::= <Central File Header Signature> <Version Made
By> <Version Needed to Extract> <General Purpose Bit Flag>
<Compression Method> <Last Mod File Time> <Last Mod File Date><CRC-
32> <Compressed Size> <Uncompressed Size> <File Name Length> <Extra
Field Length> <File Comment Length> <Disk Number Start> <Internal
File Attributes> <External File Attributes> <Relative Offset of
Local Header> <File Name> <Extra Field> <File Comment>
```

```
<Central File Header Signature> ::= 0x02014b50
```

```
<Version Made By> ::= <Integer16>
```

```
<Version Needed to Extract> ::= <Integer16>
```

```
<General Purpose Bit Flag> ::= <Integer16>
```

```
<Compression Method> ::= <Integer16>
```

```
<Last Mod File Time> ::= <Integer16>
```

```
<Last Mod File Date> ::= <Integer16>
```

```
<CRC-32> ::= <Integer32>
```

```
<Compressed Size> ::= <Integer32>
```

```
<Uncompressed Size> ::= <Integer32>
```

```
<File Name Length> ::= <Integer16>
```

```
<Extra Field Length> ::= <Integer16>
```

```
<File Comment Length> ::= <Integer16>
```

```
<Disk Number Start> ::= <Integer16>
```

```
<Internal File Attributes> ::= <Integer16>
```

```
<External File Attributes> ::= <Integer32>
```

```
<Relative Offset of Local Header> ::= <Integer32>
```

```
<File Name> ::= <octet>[File Name Length]
```

```
<Extra Field> ::= <octet>[Extra Field Length]
```

```
<File Comment> ::= <octet>[File Comment Length]
```

Digital signature:

<Digital Signature Header> ::= <Header Signature> <Size of Data>
<Signature Data>

<Header Signature> ::= 0x05054b50

<Size of Data> ::= <Integer16>

<Signature Data> ::= <octet><Size of Data>

OFFICE 2007.A.3: **END**

Zip64 End of Central Directory Record

PRODUCT: WORD, EXCEL, POWERPOINT

GRAMMAR:

< Zip64 End of Central Dir Record> ::= <Zip64 End of Central Dir
Signature> <Size of Zip64 End of Central Dir Record> <Version Made
By> <Version Needed to Extract> <Number of this Disk> <Number of
Disk with Start of Central Directory> <Total Number of Entries in
Central Dir on Disk> <Total Number of Entries in Central Directory>
<Size of Central Directory> <Offset of Start pos between Central Dir
and Disk Num> <Zip64 Extensible Data Sector>

<Zip64 End of Central Dir Signature> ::= 0x06064b50

<Size of Zip64 End of Central Dir Record> ::= <Integer64>

<Version Made By> ::= <Integer16>

<Version Needed to Extract> ::= <Integer16>

<Number of this Disk> ::= <Integer32>

<Number of Disk with Start of Central Directory> ::= <Integer32>

<Total Number of Entries in Central Dir on Disk> :: = <Integer64>

<Total Number of Entries in Central Directory> ::= <Integer64>

<Size of Central Directory> ::= <Integer64>

<Offset of Start pos between Central Dir and Disk Num> ::= <Integer64>

<Zip64 Extensible Data Sector> ::= Special Purpose Data Block

```
<Special Purpose Data Block> ::= <BlockElement>|<BlockElement<Special
    Purpose Data Block>
```

```
<BlockElement> ::= <Header ID><Data Size><Special Purpose Data[Data
    Size]>
```

OFFICE 2007.A.4: **END**

Zip64 End of Central Directory Locator

PRODUCT: WORD, EXCEL, POWERPOINT

GRAMMAR:

```
<Zip64 End of Central Directory Locator> ::= <Zip64 End of Central Dir
    Locator Sig> <Num of Disk with Start of Zip64 End of Central Dir>
    <Relative offset of Zip64 End of Central Dir Record> <Total Number
    of Disks>
```

```
<Zip64 End of Central Dir Locator Sig> ::= 0x07064b50
```

```
<Num of Disk with Start of Zip64 End of Central Dir> ::= <Integer32>
```

```
<Relative offset of Zip64 End of Central Dir Record> ::= <Integer64>
```

```
<Total Number of Disks> ::= <Integer32>
```

OFFICE 2007.A.5: **END**

End of Central Directory Record

PRODUCT: WORD, EXCEL, POWERPOINT

GRAMMAR:

```
<End of Central Directory Record> ::= <End of Central Dir Signature>
    <Number of this Disk> <Number of Disk with Start of Central Dir>
    <Total Number of Entries in Central Dir on Disk> <Total Number of
    Entries in Central Dir> <Size of the Central Directory><Offset
    between Start of Central Dir and Disk Number> <.ZIP File Comment
    Length> <.ZIP File Comment>
```

```
<End of Central Dir Signature> ::= 0x06054b50
```

```
<Number of this Disk> ::= <Integer16>
```

```
<Number of Disk with Start of Central Dir> ::= <Integer16>
```

```
<Total Number of Entries in Central Dir on Disk> ::= <Integer16>
```

```
<Total Number of Entries in Central Dir> ::= <Integer16>
```

```
<Size of the Central Directory> ::= <Integer32>
```

```
<Offset between Start of Central Dir and Disk Number> ::= <Integer32>
```

```
<.ZIP File Comment Length> ::= <Integer16>
```

```
<.ZIP File Comment> ::= <octet>[.ZIP File Comment Length]
```

OFFICE 2007.A.6: **END**

The remaining constructs are N/A.