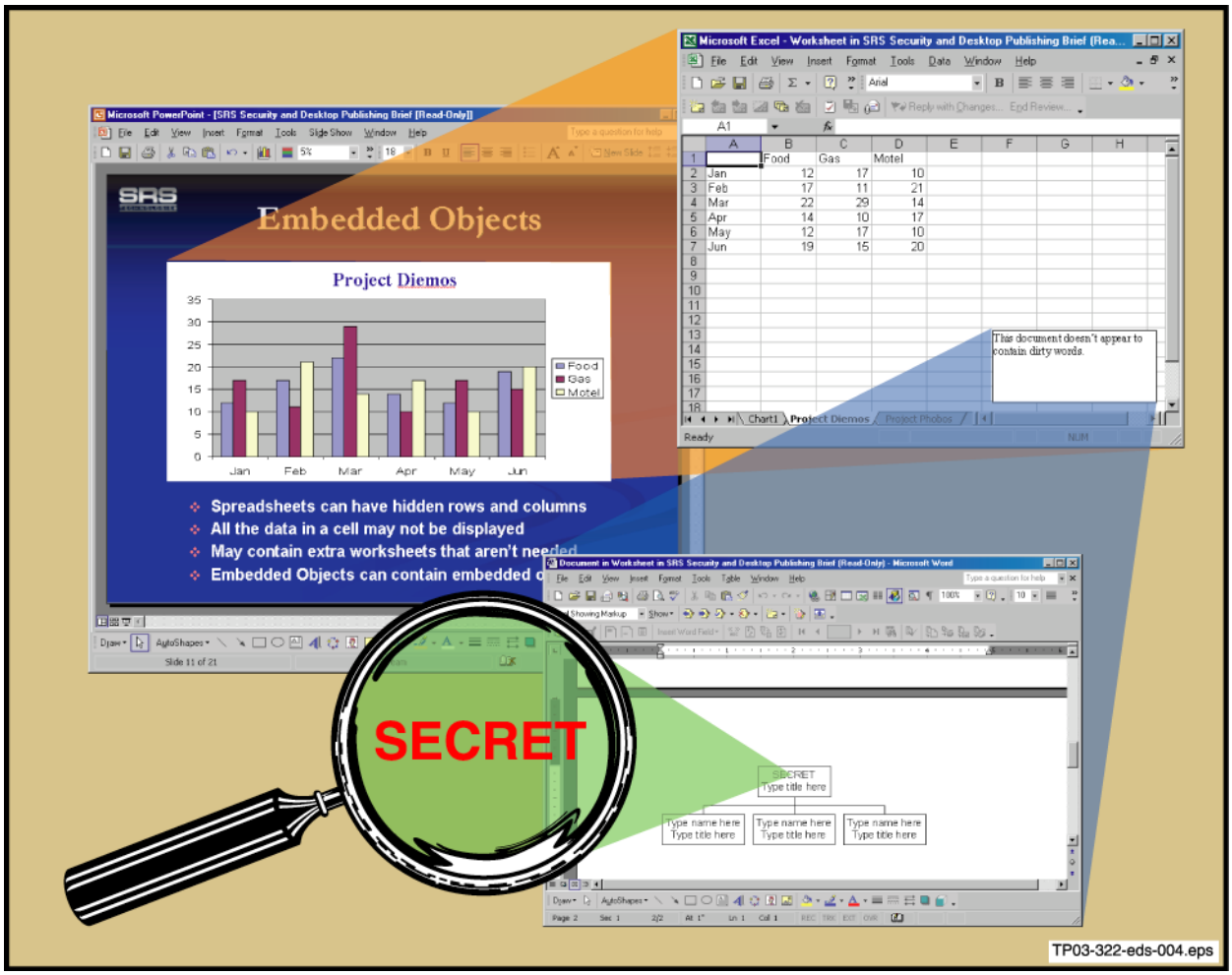
ManTech
International Corporation ®

Introduction to Electronic Document Security



December 9, 2005

By Ronald D. Hackett, PE

ManTech SRS Technologies
Systems Solutions Division
500 Discovery Drive
Huntsville, AL 35806
Phone (256) 971-7000
Fax (256) 971-7067
Ronald.Hackett@ManTech.com

Table of Contents

Table of Figures

White Paper on Electronic Document Security

1.0 Introduction

Have you noticed that incidents involving electronic documents, and especially Microsoft Office and their Tracked Changes feature are on the rise? In October 2005, the United Nations became the center of controversy when a report on the assassination of Lebanese Prime Minister inadvertently exposed the names of the suspects.[1] In September 2005, the letter sent by the United Kingdom's Home Secretary supporting new anti-terrorist measures was found to contain a deleted paragraph questioning those same measures.[2] [3] Last summer, the White House was embarrassed when their digital fingerprints were found in independent congressional testimony,[4] classified portions of the Army's redacted report on the shooting of the Italian journalist in Iraq was recovered by the Italian press,[5] and Pentagon computers were exposed to potential attack when a hacker indictment inadvertently exposed their IP addresses.[6] More recently, the hidden data in the President's "Plan for Victory in Iraq" exposed the author, raising questions about who was in charge of policy development.[7] Simon Byers from AT&T conducted a study and found that 93% of Microsoft Word documents contained hidden text.[8] You will find references to more incidents like these in Section 4 of this paper. Could this be an epidemic?

What the public does not know, and what we have not seen published anywhere, is that Microsoft automatically enables Tracked Changes. Microsoft Office XP greatly expanded the Tracked Changes feature and included special hooks in Outlook called, "Reply with Changes." Any time a Word, PowerPoint or Excel document is sent using Outlook as the email client, Microsoft automatically enables Tracked Changes without warning the user. It is quite possible that no one at the U.N., the U.K. Home Secretary's office or the White House intentionally used the Tracked Changes feature. All they had to do was email the document to someone else!

---

[1] Wait, Patience, and Onley, Dawn S., "Document security flap at U.N. causes uproar," *GCN Magazine*, 25 Oct 2005. http://www.gcn.com/vol1_no1/daily-updates/37416-1.html

[2] Sturgeon, Will, "Blunder in Word shows government terror doubts, When will they learn?," *Silicon.com*, 16 Sep 2005. http://software.silicon.com/security/0,39024655,39152367,00.htm

[3] Millman, Rene', "Expert blasts Home Secretary email blunder," *SC Magazine*, 16 Sep 2005. http://www.scmagazine.com/news/index.cfm?fuseaction=newsDetails&newsUID=333b032e-eefa-498c-b9a8-2e893cb72b0c&newsType=Latest%20News&s=n

[4] Hamburger, Tom, "Nonpartisan Testimony Gets White House Edit," *Los Angeles Times*, 19 May 2005.

[5] Jesdanum, Anick, "Military Mistake caused data leak," *Associated Press*, 2 May 2005. http://www.businessweek.com/ap/financialnews/D89R8NR80.htm?campaign_id=apn_tech_down

[6] Leyden, John, "Pentagon uber-hacker rap sheet spills attach details," *The Register*, 11 July 2005. http://www.theregister.co.uk/2005/07/11/mckinnon_indictment_snafu/

[7] Wait, Patience, "White House accidentally exposes data in PDF file", *GCN Magazine*, 5 Dec 2005. http://www.gcn.com/vol1_no1/daily-updates/37688-1.html

[8] Byers, Simon, "Information Leakage Caused by Hidden Data in Published Documents," *Security & Privacy*, Vol. 2, No. 2, pg 23-27, IEEE Computer Society, March/April 2004.

Once enabled, the Tracked Changes feature can be extremely difficult to remove. Even Microsoft's Remove Hidden Data (RHD) plug-in fails to remove the Tracked Changes from a PowerPoint presentation. Few users know to look for a "Reply with Changes" button on their Office toolbar, which indicates the Tracked Changes feature is enabled. Curiously missing is the "End Review" button that is supposed to terminate the review. In PowerPoint, only the person who started the Review can actually turn it off while the document is in a review cycle. Imagine the difficulty of finding that individual if they do not even know they turned it on.

Computers and the Internet have created a tremendous need to share information in an electronic format. It is much easier to assimilate and use information that is already in electronic form. Some data, like models, simulations, and databases, can only be used in digital form. Software applications and application suites, like Microsoft Office (MSO), permit the seamless integration of data from different applications to produce professional looking documents. This amalgamation of information into single, monolithic files is typically called *Desktop Publishing*. Desktop publishing combined with the Internet, and especially email, makes it very easy to share information. The Department of Defense has also capitalized on these information sharing capabilities with Internet-like networks like Intelink and the SECRET Internet Protocol Routed Network (SIPRNET). Everyone knows that sharing information across secure boundaries is a risk, but the full extent of the risk is not well-known.

Sensitive and classified information is routinely and unwittingly compromised by hidden data in desktop publishing documents. Computer-generated documents and files often contain hidden information that is unknown to authors and readers, but could be exploited by knowledgeable third parties. Keyword scanners used to screen information are inadequate by themselves, compounding the problem and significantly increasing the risk. Other commercial software packages like *Protect* from Workshare, *Metadata Assistant* from Payne Consulting, and *ezClean* from KKL Software may be good tools for some commercial applications, but they are not a complete document security solution by themselves and lack sufficient rigor to meet statutory and regulatory requirements for protecting classified information. They may even exacerbate the problem by removing tags that a keyword scanner would catch.

SRS has developed technology that will allow the software application user to identify and control the excess information that can become embedded in an electronic application file without the user’s knowledge or direct consent. This technology provides both a software product that will address commercial and consumer security needs, and a service that will address high security needs of protecting National Security Information. We call this new discipline Electronic Document Security (EDS). It is a highly specialized subset of Information Security (IS) that focuses on the contents of electronic documents, and not malicious actions by people. Traditional IS focuses more on the hacker or the malicious insider who is intent on stealing or damaging information. Our society spends millions of dollars protecting information from hackers and malicious insiders while spending almost nothing to prevent sensitive information from leaking out in legitimate and routine electronic document exchanges. Ironically, the biggest threat to sensitive information may be the honest user just doing their job.

2.0 Background

Microsoft's Object Linking and Embedding (OLE) and Component Object Model (COM) standards permit seamless integration of software applications to produce professional looking documents commonly described as *Desktop Publishing*. Unfortunately, these standards do not consider privacy or security, leading to significant vulnerabilities. There are at least four serious vulnerabilities that must be considered in any digital document security program. These four areas are embedded objects, Meta data, file fragments, and highly formatted information.

2.1 The Embedded Object Threat

Embedded objects contain all of the data contained in the original file, but only display a small portion of the information. As a result, the user is misled regarding the amount of data actually contained in the file. The problem is aggravated by the ability to "crop" and "resize" embedded objects. Cropping and resizing are misnomers that imply information has been discarded. Only the top-level presentation is changed. The embedded objects still contain all of the original information. **Figure 1** shows an image that has been cropped to hide the simulated classified axes labels. **Figure 2** shows the same image uncropped to reveal the simulated classified axes that were not intended for sharing. Embedded objects can even contain other embedded objects, further compounding the problem of hidden data. There is no theoretical limit to the number of nested embedded objects that can be contained in a document.

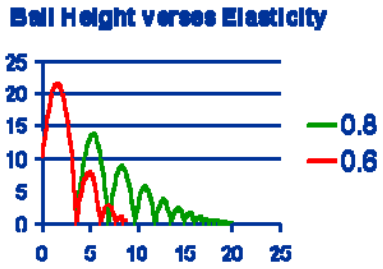


Figure 1, Cropped Image Hiding Simulated Classified Axes Labels

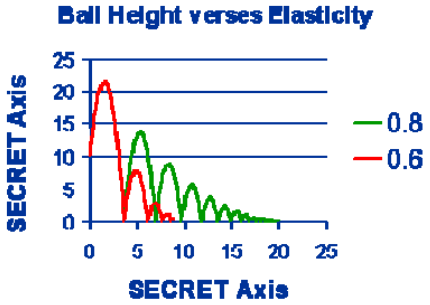


Figure 2, Image Uncropped to Reveal Simulated Classified Axes Labels

A quick example of the embedded object problem is shown in **Figure 3**. This Figure shows a screen shot of a Powerpoint 2002 presentation that contains an embedded Excel chart. This Figure clearly shows the keyword 'secret' in the legend of the chart; however, when the file is reviewed in a binary editor or with a keyword scanner, the keyword cannot be located. In this situation, only the human reviewer can see the keyword. **Figure 4** shows another screen shot of the same Powerpoint file, but this time the legend is turned off. The keyword 'secret' still appears in the second worksheet that contains the data displayed on the graph, but the information is not transferred to the worksheet that displays the chart. In this case, the keyword is not obvious to the human reviewer, and the keyword cannot be located in the file using a text scanner. Unless the human reviewer is highly trained, they will likely miss the hidden information.

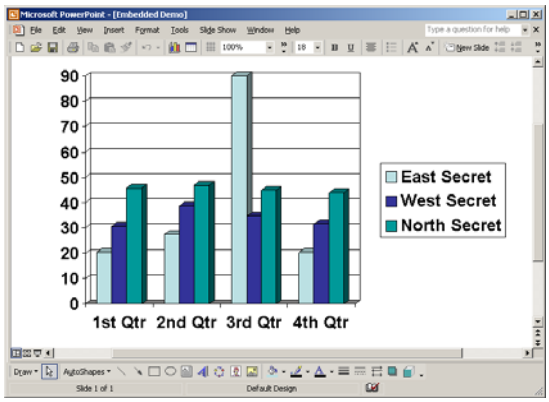


Figure 3, Embedded Excel Chart and Simulated Secret Legend Visible

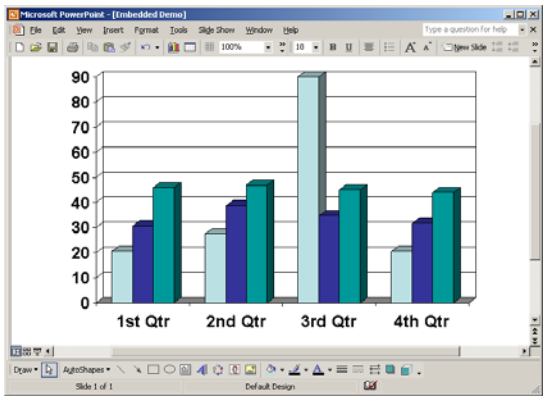


Figure 4, Embedded Excel Chart with Simulated Secret Legend Suppressed

The placement of embedded objects can also cause problems. Beginning in Office 97, Microsoft removed the page boundaries that restricted authors from dragging objects off the page. This can be helpful when working in a cluttered environment, such as a busy PowerPoint slide, but we have seen cases were a user has dragged an object off the page and forgotten to retrieve it. **Figure 5** shows an image of a U. S. Air Force B-2 bomber that was dragged off the bottom of a slide. The image can be seen by carefully manipulating the scroll bar. The image is not visible when paging through the presentation. The same capability exists in Word, but the image is not displayed in any view! **Figure 6** shows the same B-2 image dragged half way off the page. Notice that the portion that is off the page cannot be viewed. When the entire image is dragged off the page, it can only be located by "fishing" around with the mouse. It is hard to find the object even when you know where it should be located. This does not prevent a knowledgeable analyst from recovering the image.

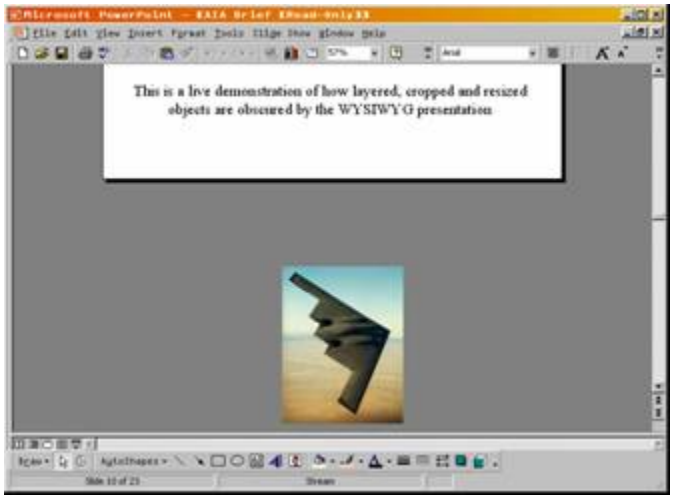


Figure 5, Off Page B-2 Image in PowerPoint

2.2 The Meta Data Threat

Another significant problem is the extensive use of Meta data in an OLE/COM file. Meta data is called “hidden” data because the user is usually unaware that this data is present. Meta data helped incriminate the developer of the Melissa macro virus. The *Wall Street Journal* talked about the problem of Meta data in a 20 October 2000 article by Michael McCarthy. This article cites one example of political staffers sending what they thought were “anonymous” documents to their rivals. The perpetrators were clearly identified in the Meta data. A White Paper by Workshare Technologies reports an instance where the British Government published a Word document with the Track Changes option enabled to its web site. The tracked changes revealed how politically sensitive paragraphs had been reworded. The “leak” was widely reported in the media, which lead to “damaging accusations” against the Government. There are

many different types of Meta data in a file. As an example, Microsoft Office can keep track of the last 10 people that worked on a file. Microsoft also keeps track of the original location and file name of an embedded object. File names can be very descriptive, so it is possible an unlabeled or generically labeled image or diagram is specifically identified in the Meta data.

To demonstrate the Meta data problem, ManTech SRS Technologies created a simple Microsoft Word 2002 document that contains a single line of text. The text reads, "This document contains the dirty word SECRET." After the document has been saved under the name, "Word Demo," the text is changed to read, "This document contains no dirty words." The corrected document is then saved and closed. Next, the file itself is examined using a binary editor, or in this case, Notepad. A screen shot of the file in Notepad is shown in **Figure 7**. The corrected, or sanitized, text is highlighted with a purple circle (Note 1). The original, unsanitized text is also visible three times in the file and is highlighted with red circles (Note 2). The author and the company are also identified in the file. This information is highlighted with blue circles (Note 3). The fully qualified file name, which includes the complete path information, is also recorded in the file and is highlighted with a green circle (Note 4). Users usually take advantage of long file names and give their files very descriptive names. If the user later decides to give the file a more generic name, or if the user uses this file as a template for a new document, the original file name remains embedded in the file. This embedded file name could reveal information the user did not intend to share. All of this information is automatically collected and stored as part of the file without any user action or intervention. Note that some binary data has been removed from this file to make this view more presentable to the human eye. In its native state, the information is not as tightly grouped as shown in Figure 7. Some of the information in Figure 7 is accessible through the document properties window, shown in **Figure 8**, but most of the information is beyond the control of the user. Warning! Attempting to use either Notepad or a binary editor to excise this hidden data will likely damage the digital file and make it inaccessible to its native application.

Figure 6, Off Page B-2 Image in Word

Many organizations rely on the user to review documents before they are shared with others outside the organization, but the user is rarely trained in how to review a document. Most users rely on the "What You See Is What You Get" (WYSIWYG) presentation to review a document. WYSIWYG (pronounced wiz-e-wig) is a programming concept built into the application that makes the information displayed on the computer monitor look the same as the printed document. While it is convenient, WYSIWYG leads the user into thinking in terms of the one dimensional printed document instead of the multidimensional electronic document. Unfortunately, WYSIWYG only displays a small portion of the information actually contained in

the file. Even a highly trained user will have trouble reviewing a document because of the plethora of features built into various applications.

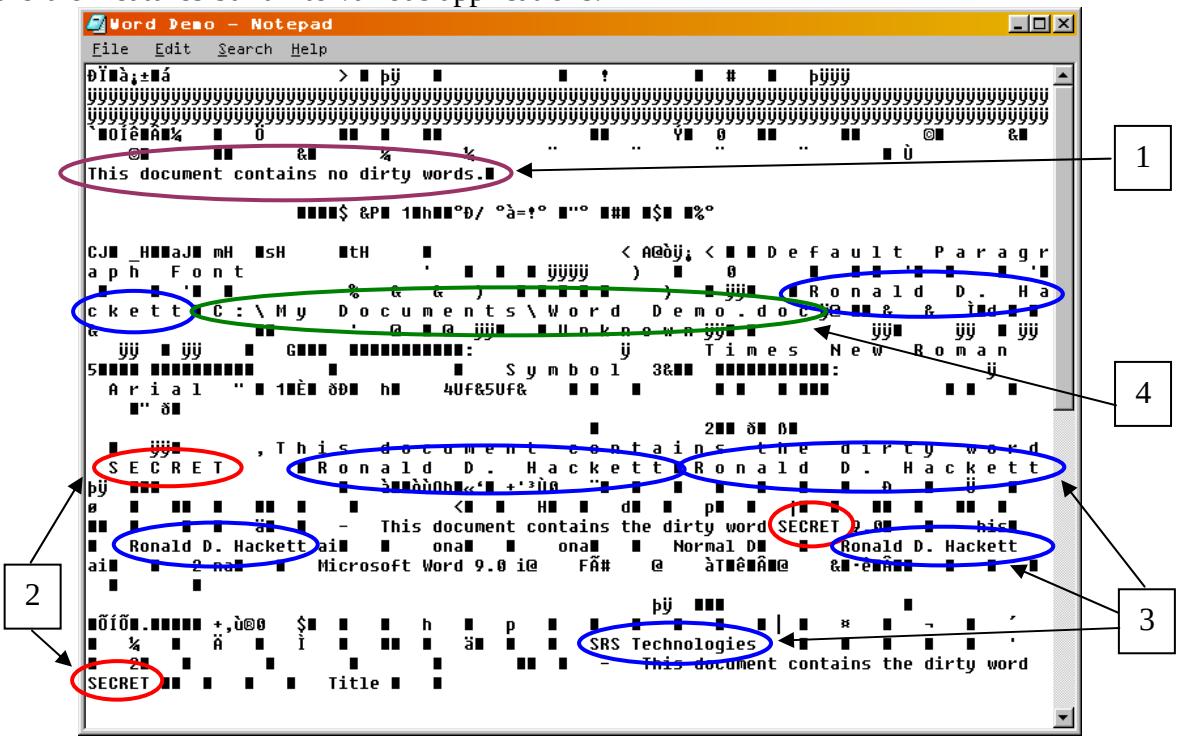


Figure 7, Notepad View of Demo Document with Meta Data Highlighted

For several reasons, keyword scanners, euphemistically called “dirty word” scanners, frequently used to screen documents prior to release are not adequate and give users a false sense of security. These scanners assume that all information is stored in a known format, typically ASCII or Unicode. Many file types are moving toward proprietary formats to protect content. Adobe’s Portable Document Format (PDF) is one prominent example. The reader software is freely available for viewing the information, but it is difficult to extract information without buying the editor software. Compression is also becoming prevalent because of the need to share information over networks like the Internet. In many cases this compression is internal to the document and not obvious. Microsoft Office 2000/2002 uses compression to minimize the file size, which makes parts of the document unreadable to a keyword scanner. Keyword scanners also depend on the sensitive information being properly marked, but experience shows many digital documents are not subject to the same standards and scrutiny as printed documents. Even when the content is marked according to security guides, the Meta data is not.

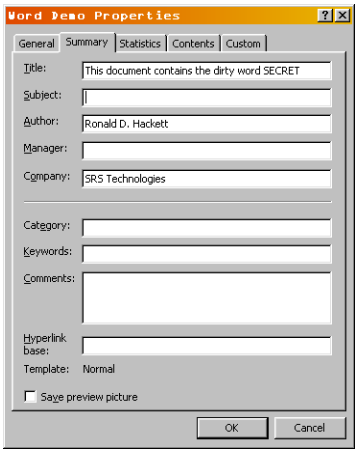


Figure 8, Document Properties

2.3 The File Fragmentation Threat

A third vulnerability caused by the OLE/COM file structure is file fragmentation. OLE/COM files are extremely complex and consist of multiple data streams grouped into storages. Storages can contain any number of data streams, and they can also contain other storages. This is analogous to the directory structure of the file system, where storages are the equivalent of directories and streams are the equivalent of files. Microsoft actually describes OLE/COM files as, "a file system within a file." The hierarchical structure of a Word document is shown in **Figure 9** using the *DocFile Viewer* utility that is included with *Microsoft Visual C++*. The root document appears as a folder at the top of the viewer, and the six data streams that comprise the electronic document are shown inside the folder. In this Figure, the *Document Summary Information* (Meta data) is highlighted. When an embedded object, such as an Excel Workbook, is inserted into the document, the root storage of the workbook is inserted into the Word document's root storage as a substorage. This is shown in **Figure 10**. An intermediate storage (or folder) called the *ObjectPool* is created to group similar objects together even though there is only one object in the collection. The embedded Excel workbook and the Meta data for both the document and the workbook are indicated in the Figure.

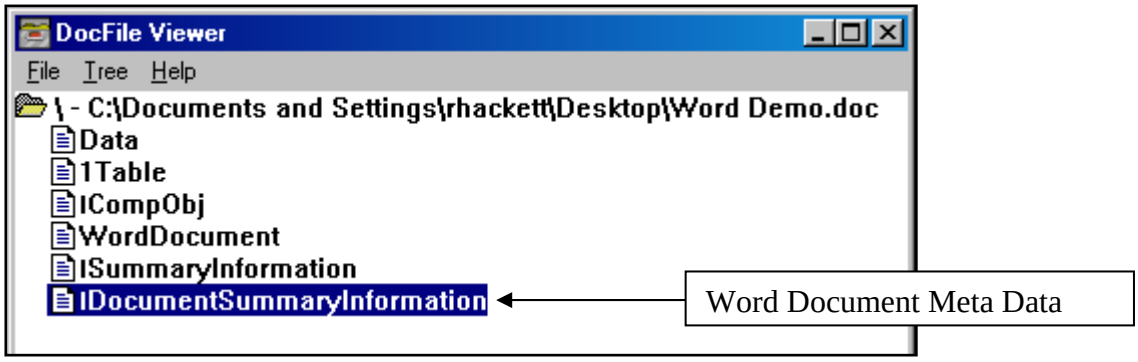


Figure 9, Streams in a Simple Word Document

OLE/COM files can suffer from fragmentation and slack space just like a file system. Many people are now aware that deleting a file from the file system does not remove the information from the disk. Similarly, deleting information from a document does not necessarily remove the information from the file. The fragmented space in an OLE/COM file can contain information that has been "deleted" from the document. This is the reason files always seem to grow larger, even when information has been 'deleted' from the document.

The problem of fragmented files is more difficult to demonstrate. **Figure 11** was taken from the Microsoft OLE/COM documentation. It shows graphically how a fragmented file can be defragmented using the 'CopyTo' API. In this Figure, the fragmented file is shown at the top. Streams can be non-contiguous and the file contains 'unused' space. This again is a misnomer. The 'unused' space is really 'formerly used' space that may still contain traces of the original information. The 'CopyTo' API can be used to reassemble the data into contiguous streams and remove the 'unused' space. The end result is a smaller file. Microsoft claims to have fixed this vulnerability in Windows 2000 and XP, but there are still many questions about how they handle legacy documents, embedded objects, and more complex documents.

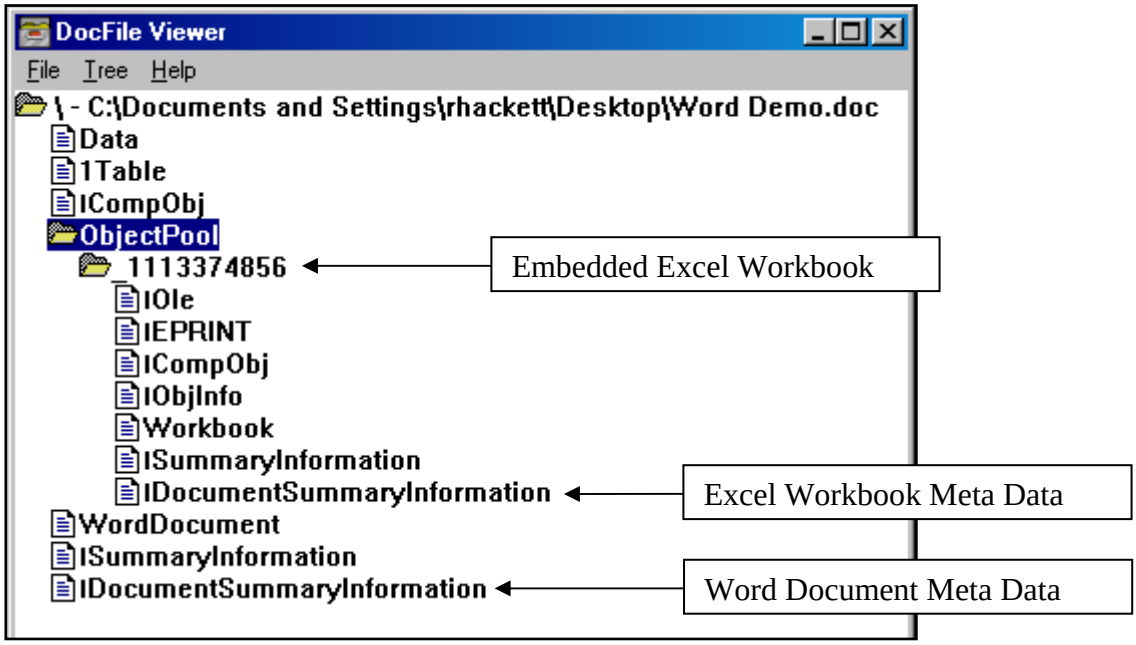


Figure 10, Word Document with Embedded Excel Workbook

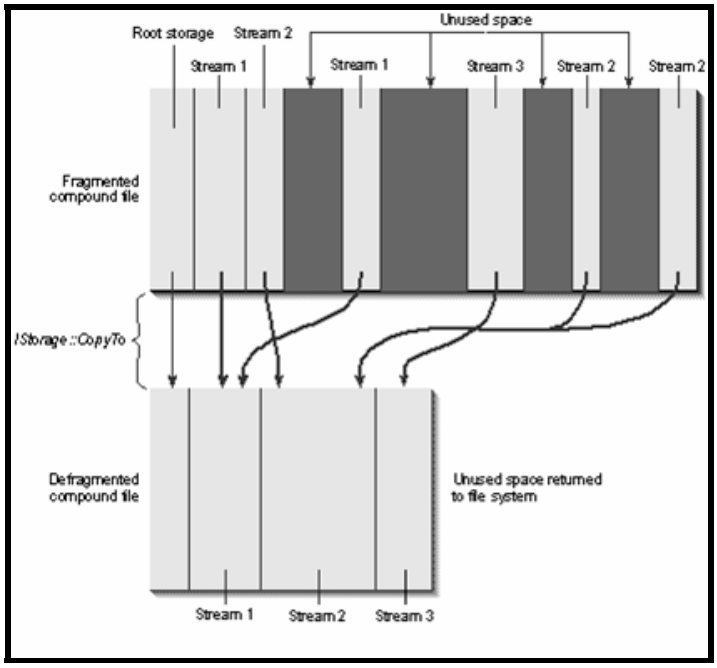


Figure 11, MSDN Graphic Showing How Defragmentation Process Works

2.4 Microsoft Office XP Threats

Each edition of Microsoft Office has added features that make the new version even more dangerous than previous versions. It's ironic that two of the new XP vulnerabilities we will discuss here were actually intended to reduce vulnerabilities of earlier versions. The general Microsoft approach is a quick patch to address the obvious issues, but they do not do the research necessary to fully address all of the potential ramifications of a patch.

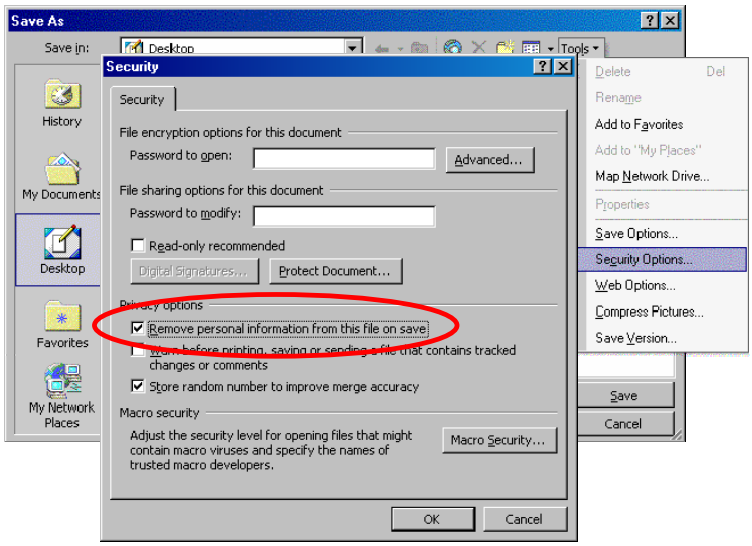


Figure 12, Accessing Microsoft Office XP Privacy Switch

The first feature was intended to address user concerns about Meta data and privacy. Microsoft added a privacy switch that is accessible from the 'Tools' tab on the 'Save As' dialog. Selecting the 'Security Options' feature brings up the Security menu as shown in **Figure 12**. Checking the box next to the "Remove personal information from this file on save" does appear to work as advertised on simple documents, but it quickly fails on more complex documents. While the switch does remove some information, it is dangerous because it gives the user a false sense of security. If the user falsely believes the problem has been alleviated as indicated by Microsoft, then they will be less attentive in reviewing the information for transfer.

Another new feature added in Office XP is the 'Compress Pictures' capability. Again, this feature is available under the 'Tools' tab of the 'Save As' dialog. It is also available on the 'Picture' toolbar. Selecting the 'Compress Pictures' feature from the menu brings up the dialog shown in **Figure 13**. The user is presented with options that appear to allow them to compress their images and to delete the cropped areas of images. Primarily intended to help users reduce the size of very large documents, especially PowerPoint

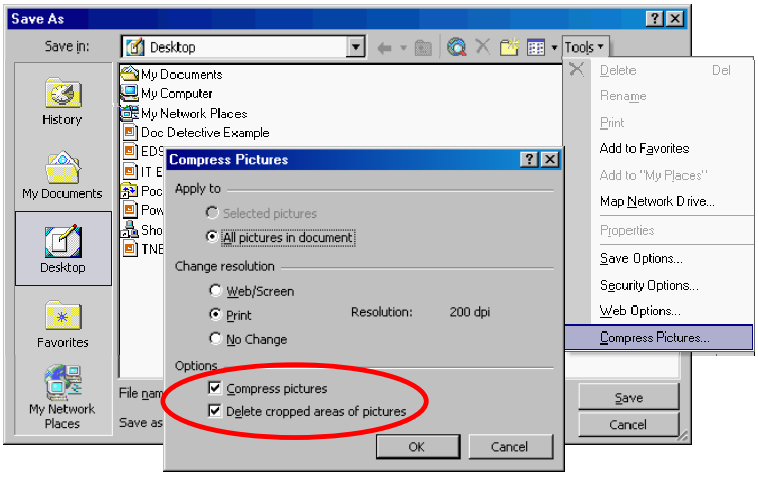


Figure 13, Accessing Microsoft Office XP Compress Pictures Feature

presentations, the feature would have security benefits as well. Unfortunately, this option fails under a number of scenarios, and the user is not warned when there is a failure. Again, a false sense of security negates any benefits gained from having this option available.

One of the most damaging features added to Office XP is an extension of the 'Track Changes' capability of earlier versions. When 'Reviewing' is turned on, complete copies of each edition of the document are stored in the file. **Figure 14** shows a one slide PowerPoint presentation that had the 'Reviewing' feature enabled. In this view, the current version of the slide is shown along with all three previous versions. Each user deleted all the information on the slide and replaced it with their own information, but all of the changes were recorded. The user can restore any previous version simply by clicking on it.

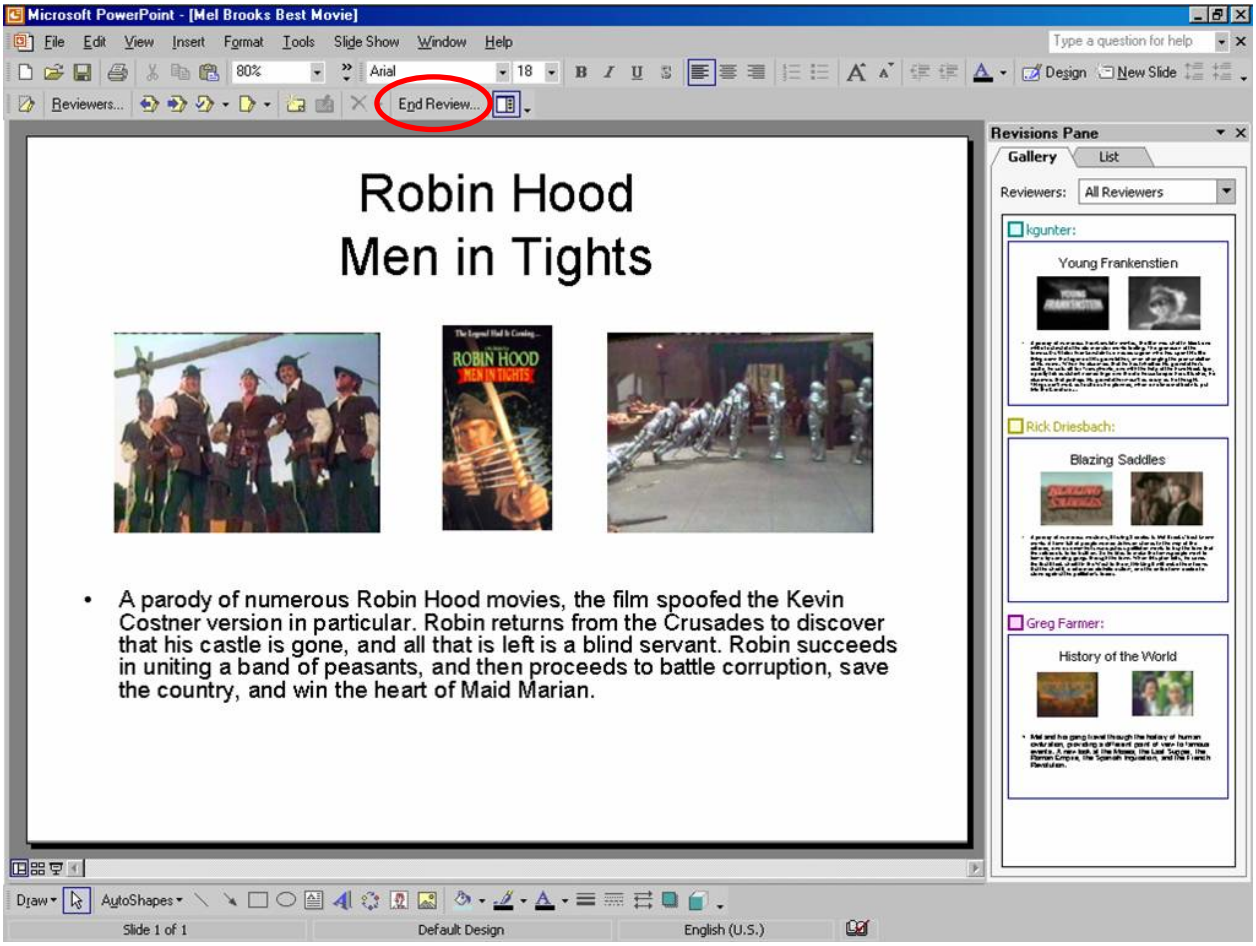


Figure 14, Microsoft Office XP Reviewing Feature (Not The Default View)

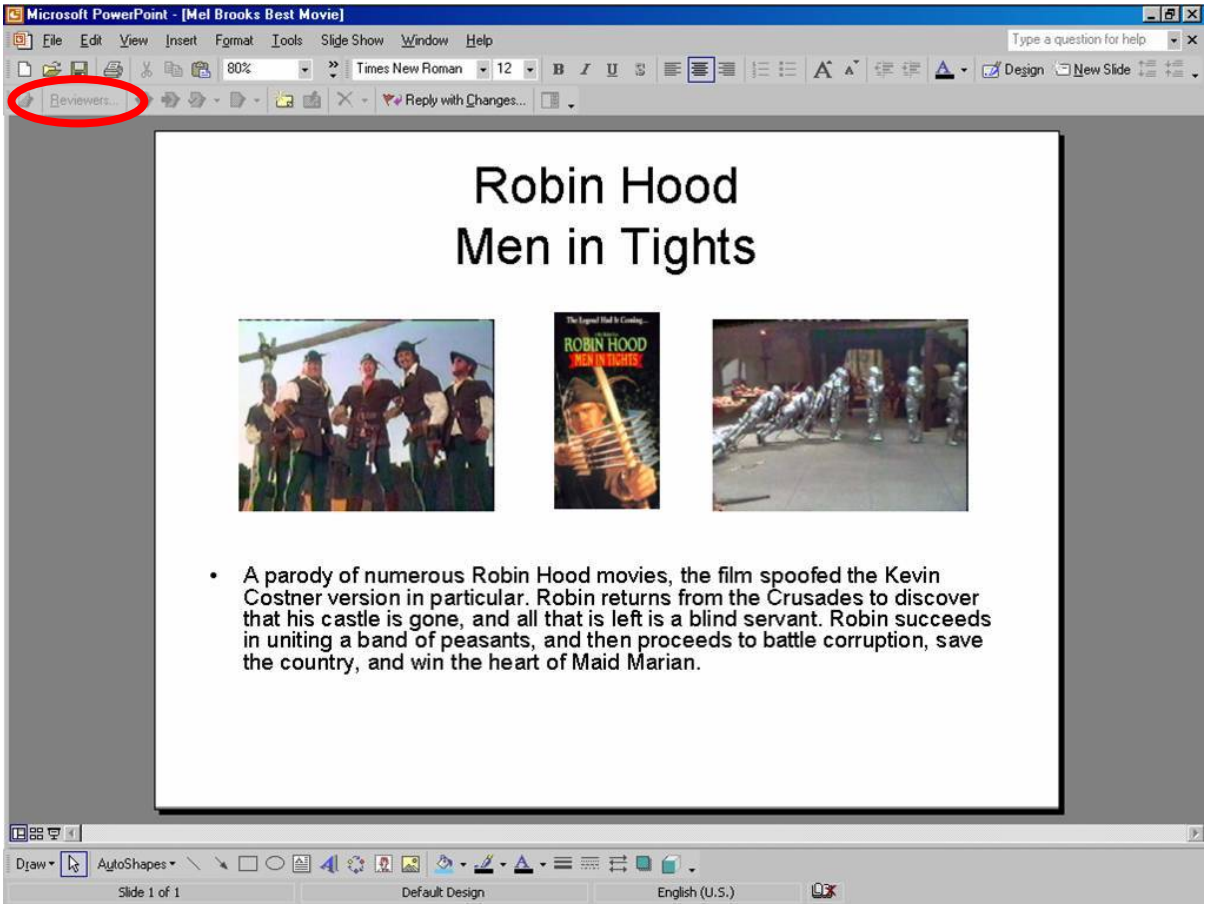


Figure 15, Misleading Reviewer Display When XP Reviewing Is Enabled

While this seems to be a useful feature, it can be turned on automatically without the user's knowledge when emailing a document using Microsoft Outlook. The default view is not the one shown in Figure 14, and many users remain unaware that the feature has been enabled. Only the user who enabled this feature can turn it off by clicking on the 'End Review' button on the 'Reviewing' toolbar. All other users who edit the file will see the presentation as shown in **Figure 15**, which does not appear to have reviewing enabled.

Because the reviewing feature captures complete copies of each revision of a document, the file size quickly escalates. Users soon notice that email systems choke on the very large files that result, but still they have no idea what causes the files to grow so rapidly. Users who receive a document that already has "Reviewing Enabled" are powerless to turn it off. **Figure 16** shows the results of the demonstration file size as it was emailed around the network. The original file was

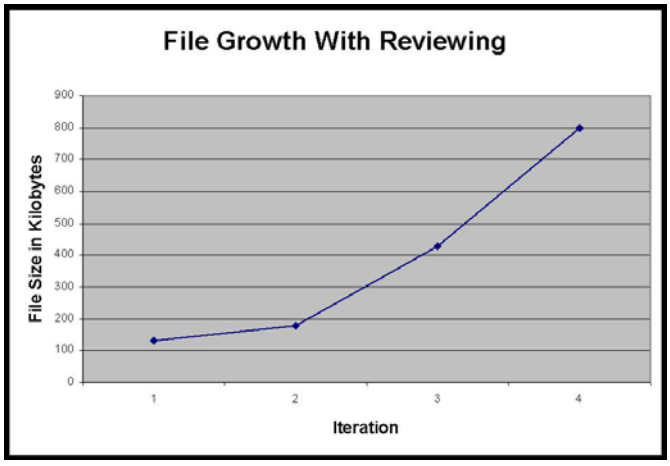


Figure 16, File Growth When XP Reviewing Feature is Enabled

138 kilobytes. After three reviewers, the file had grown to 799 kilobytes. Figure 16 clearly shows the exponential file growth associated with this feature. SRS has already developed a capability to disable the reviewing feature and remove the reviewing data even when we did not enable the feature.

In response to pressure from the Government, Microsoft developed and released a Remove Hidden Data (RHD) plug-in for Word, PowerPoint, and Excel. Unfortunately, this plug-in does not do the job very well. One problem is their failure to remove the Reviewing data that has been collected and stored in a PowerPoint file. The RHD Plug-in reports that Reviewing RCIDs were found and removed, but the reviewing data remains in the file. Using the technology developed by SRS Technologies, we can recover that data.

2.5 The Highly Formatted Information Threat

Hyper Text Markup Language (HTML) documents have a reputation in the security community as relatively "safe" documents because they are text based documents that can be reviewed with simple tools. That may have been true in the early days of the World Wide Web, but that is no longer true of today's state-of-the-art HTML documents and the related eXtensible Markup Language (XML) documents. These documents contain cascading style sheets, conditional links, tables and other highly formatted data structures that are difficult for users to analyze. Conditional formatting allows the structure of the page to change depending on the browser in use. All of this formatting makes it impossible for most users to reliably review an HTML or XML document. Try reviewing an Excel spreadsheet that has been exported to a web page in a text editor like Notepad, and you will have to go several pages into the document before you find the first displayed information. If the spreadsheet contains hidden rows or columns, those rows and columns will be in the new web document, but they will not display on the browser. We have also seen text formatting errors, such as non-viewable formatting information embedded inside a word that would prevent both a human reviewer and a typical keyword scanner from detecting an obvious keyword like SECRET.

The Department of Defense published a web document called "DOD 101" (**Figure 17**) that is an excellent example of common practices with electronic documents that are dangerous to security. The document was created by exporting a PowerPoint presentation using Microsoft's "Save As Web Page" feature. The files created using this process include a filelist.xml and several *.mso and *.wm? files that are conditionally linked to the web document. The original presentation can be completely reproduced by typing the URL into the file open dialog in PowerPoint. The resulting document is nearly identical

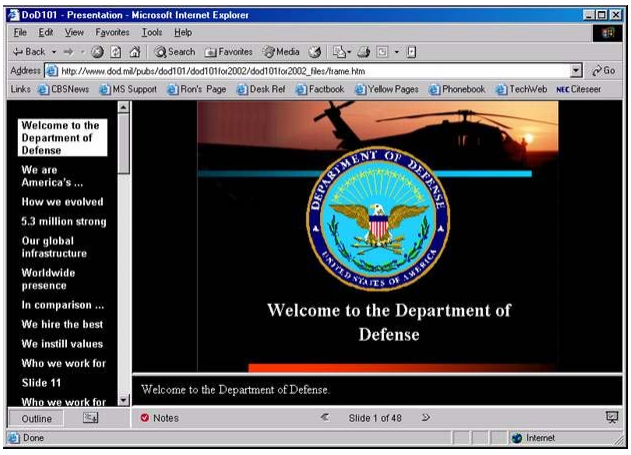


Figure 17, DOD 101 Web Page

to the original. The original file was also available on the website, so it was downloaded for comparison. The original file is slightly larger than the reconstructed file, but the difference can be attributed to file fragments in the original.

The background used for this presentation was obtained from "dphilbin" at "AFIS" in the form of a PowerPoint presentation. Both the user and the organization are identified in the embedded presentation's Meta data. Five copies of that presentation are embedded in the DOD 101 presentation. An image on slide 42, titled "Progress in Afghanistan," contains a hidden paragraph. The paragraph is contained in a special 'Alternative Text' field that Microsoft included to assist with the web page export feature (see **Figure 18**). Images on a web page will sometimes have an alternate text attribute assigned to them which is displayed while the image is loading or if the image is not available. These are short descriptive names because they must fit into the area allocated for the image. The alternative text assigned to this image appears to be a complete news article, and it will not fit in the space allocated to this image.

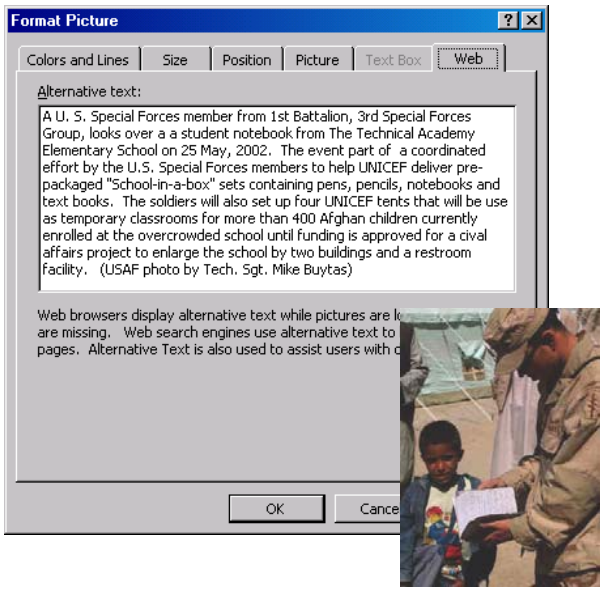


Figure 18, Alternate Text for Inset Image

Several slides contain empty text fields which are not visible on the display. There are two problems with these fields that make them dangerous to security. The fields could have contained "white text," or text that is the same color as the background, and the display would have been the same. Microsoft also permits these text fields to have alternative text, which would not be displayed.

Finally, the file size is huge. Even with the fragments and embedded presentation removed, the file is nearly 15 megabytes. This presentation contains many embedded images that have been resized and cropped, but all of the original data is still contained in the presentation. With so much detail available in the images, an adversary could obtain sensitive information by doing some image processing. After optimizing only half of the slides in the presentation, the file size was reduced to less than 4 megabytes. Optimizing the remaining slides could easily get the presentation down below 2 megabytes.

While SRS did not find any compromising material in the DOD 101 web page, the document is representative of how electronic documents are being processed in the DOD. Given the number of electronic documents made public by the DOD, it is not unreasonable to assume that some documents do compromise sensitive information.

We offer one final note about Rich Text Files (RTF), which also have an underserved reputation as "safe" documents. Many people incorrectly believe that an RTF file is a fancy text file format created by Microsoft. RTF is a fully OLE compliant file that is capable of reproducing most of the problems documented in this proposal. In fact, RTF is really a Word 2.0 document.

2.7 The Image Analysis Threat

Embedded images are popular in electronic documents. Applications have powerful utilities for adjusting the image to meet the needs of the document author, but few authors realize that the basic image is not changed by these utilities. A complete copy of the original image is stored in the electronic document when it is first inserted. Subsequent alterations to the display of the image are recorded separately in the electronic document and the altered image is recreated each time the file is opened. Extracting the original image is possible, and it can reveal information the document author did not intend to share.

Figure 19, Embedded Image Example

The image in **Figure 19** was taken from a corporate presentation found at SRS. The image is shown as it appeared in the presentation. The image had been resized to less than 30 percent of its original size and was included as an example of a computer processing facility. The original image was a 1416 x 1077 (1,525,032) pixel image which could be extracted and reprocessed to reveal additional information. **Figure 20** shows the logo on the far wall between the two large screen projections. Although it is a little fuzzy, careful analysis determined it is the National Missile Defense (NMD) logo. The logo is more legible on the computer monitor than on the printed image because of the anti-aliasing filter used in the video driver. This dates the image because NMD was renamed the Ground based Midcourse Defense (GMD) in early 2002. Careful analysis of the two large screen projections revealed that the screen on the left was displaying a missile tracking map of the world, and the screen on the right was displaying a graph that turned out to be the mission timeline. Both large screen projections had headers and footers that are consistent with classified material. The headers and footers appeared to be a long single word, which would be consistent with UNCLASSIFIED. We later identified this picture and the facility and confirmed that this was an unclassified missile tracking scenario. Three computer monitors are also visible in this photograph, but they appear to be too overexposed to recover any information. Good photo analysis techniques should not be underestimated, but none were attempted in this exercise.

Figure 20, Logo from Embedded Image

This analysis of a resized photograph in a presentation did not reveal any sensitive information, but it does indicate the potential for such information to pass undetected by the electronic document author. Newer digital cameras have improved the resolution of digital images by over a factor of three, and this trend is expected to continue. Improvements in the resolution of the images and in digital image processing techniques will enhance the ability to recover unintentional information from embedded images.

2.8 The Adobe PDF Threat

The Adobe Portable Document Format (PDF) is one of the most well known and used document formats available today. One reason for this popularity is the belief that PDF files are safe files that only contain a snapshot of the viewable portion of the original document. Such assumptions are common because most PDF writers work as print drivers, and one-dimensional printed documents are considered safe. Nothing could be further from the truth.

If an Excel spreadsheet with hidden rows and columns is ported to a PDF file, the PDF file will contain the hidden rows and columns even though those rows and columns do not appear on the printer. The Title and Author fields from the Document Summary Information (Meta data) also appear in the new document, but not on the printer. Much more information is passing through the print driver and being captured in the document than is commonly believed.

PDF documents are an object oriented collection of information. Objects are layered onto a page, just like in a Microsoft Office document. Objects can obscure other objects, and even text can be obscured. One popular method for redacting documents is to draw black boxes over the text to be redacted. Because the box and the text are different objects, the text is easily recovered. This is how the Italian press recovered classified information from a US Army document.[9] The Justice Department[10], the New York Times[11], and the Washington Post[12] have also exposed sensitive information using this redaction technique. We have also seen other cases of lost text. The top of **Figure 21** shows a section header for the Adobe Encapsulated Post Script (EPS) specification. During our research, we discovered hidden text behind the header. The bottom of Figure 21 shows the same document with the header and a white rectangle removed. **Figure 22** shows an SRS document before (left) and after (right) an object was removed. Clearly, there are hidden objects on this page that include hidden text.

Fragmentation is also a problem in PDF documents. During our research, we discovered many documents that contained deleted objects and pages. The "deleted" objects and pages can be easily recovered. When the object or page is “deleted” by the user, a new page or page tree object is created to reference the remaining objects, but the old objects are not removed. This gives the appearance that the information has been removed, but the file size does not decrease.

---

[9] Wait, Patience and Onley, Dawn, "Army sets new policy for redacted documents," *GCN*, Vol. 24 No. 32, 7 Nov 2005. http://www.gcn.com/24_32/dodcomputing/37448-1.html

[10] Poulsen, Keven, "Justice e-sensorship gaffe sparks controversy", *Security Focus*, 22 October 2003. http://www.securityfocus.com/news/7272

[11] Foss, Kurt, "PDF Secrets Revealed: PDF file redaction snafu exposes agents' identities," *Planet PDF*. http://www.planetpdf.com/mainpage.asp?webpageid=808

[12] Foss, Kurt, "Washington Post's scanned-to-PDF Sniper Letter More Revealing Than Intended," *Planet PDF*, 26 October 2002. http://www.planetpdf.com/mainpage.asp?webpageid=2434

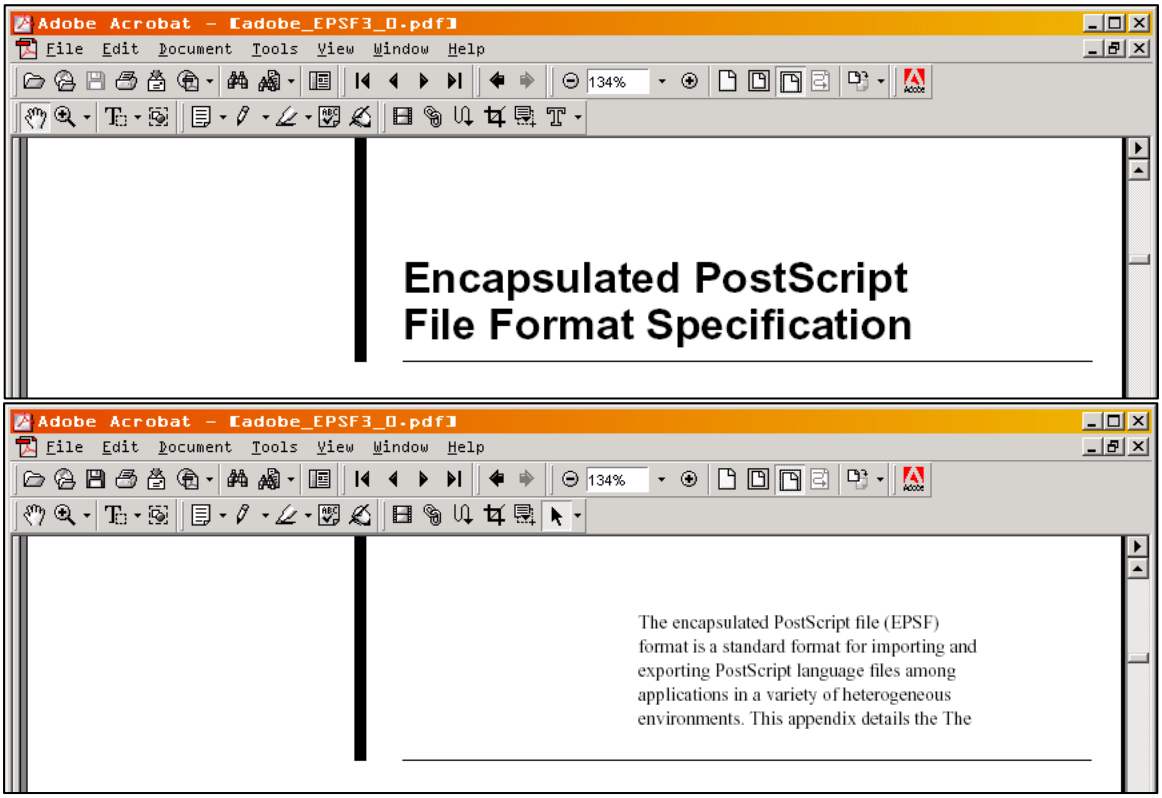


Figure 21, Hidden text (top) exposed (bottom) in the Adobe EPS specification

The PDF specification is 1,236 pages long, which indicates the complexity of the format. It contains an extremely functional drawing layer, Annotations (similar to Microsoft Comments), Articles (similar to Microsoft TextBoxes), forms, and other features that can obscure data. Multimedia files can be embedded, and the functionality and contents of the file can be expanded using third party plug-ins. One new third party plug-in prompted PC Magazine to ask, "Are your PDFs spying on you?"[13] New features have been added as the Adobe format has evolved over many years, but old features have not been removed to support backward compatibility. This mix of old and new features provides lots of opportunities for hidden data, especially in older PDF documents that have been revised by later versions of the software. In spite of its widely-held reputation, PDF should not be considered a safe file format!

[13] Fluckinger, Don, "Are Your PDFs Spying On You?," *PC Magazine*, 28 June 2005.
http://www.pcmag.com/article2/0,1759,1823029,00.asp

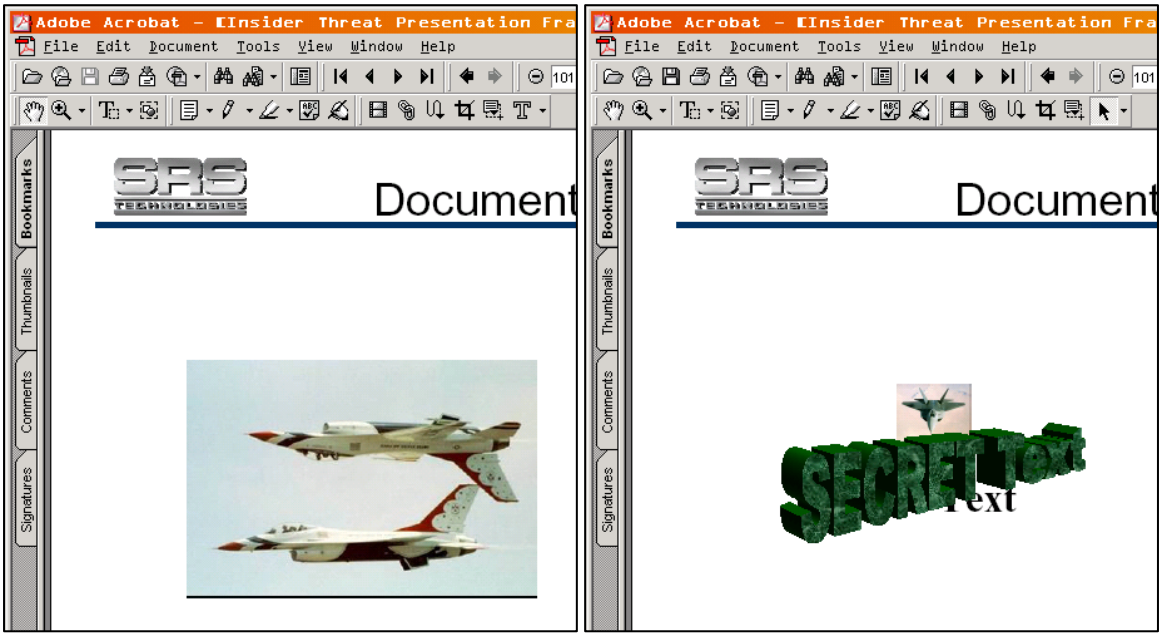


Figure 22, Layered objects in an Adobe PDF document

2.9 Keyword Scanner Limitations

Many organizations that need to transfer sensitive or classified information rely on keyword scanners to check for markings that delineate this information. Keyword scanners are sometimes euphemistically called "dirty word" scanners. Keyword searches assume that the data is being stored in a known format, typically ASCII or Unicode, and that the user has properly marked the document. The assumption that the file is stored in an ASCII or Unicode format may have been valid five years ago, but today many applications are moving toward proprietary formats to protect data. The most visible example of this trend is Adobe's Portable Document Format, or PDF. Anyone can view a PDF with a free reader, but to extract or manipulate information you need to purchase the editor. PDFs also use compression to make transmission via the Internet faster. The combination of proprietary formats and compression are files that cannot be reviewed with an ordinary keyword scanner. In this case, the keyword scanner can give a false sense of security, which exacerbates the problem.

Because keyword scanners are usually searching for text data, they are ineffective for reviewing binary information like images and drawings. This requires the user to review carefully each non-text object in a compound file. Frequently, the line between text and non-text objects is difficult for the user to determine. Sometimes the text is really an image of a printed document. This is especially prevalent with PDFs. Now the keyword scanner cannot review the information, but the user may be less rigorous in their review because they believe the keyword scanner is reviewing that information.

3.0 *Document Detective*: The SRS Solution

On 6 April 2005, ManTech SRS Technologies released a new Information Assurance software product designed from the ground up to meet rigorous U. S. Government security requirements to protect National Security Information from compromise by hidden data in

electronic documents. Specifically, it was designed to help the user conduct a "100% reliable review" of an electronic document. Because this product was designed to meet stringent Government standards, this product will be useful for business and commercial applications where the protection of a company's proprietary information and a client's private and personal data is of paramount importance. This product is fully capable of protecting the most sensitive information for the U. S. Government, so it exceeds all commercial expectations and will outperform all other similar programs currently available on the commercial market. Our product is intended to help the honest user to do their job better with fewer mistakes that could compromise security. This product is called ***Document Detective***.

3.1 PowerPoint Reviews

Document Detective takes several seconds to many minutes to analyze an electronic document. The time depends on the size and construction of the document. The median processing time is usually about one minute. When the analysis is complete, the contents of the document are displayed in a two window document browser shown in **Figure 23**. The document browser is similar to the familiar Microsoft File Explorer except that the display represents the contents of the electronic document instead of the file system. The tree window on the left contains hierarchical collections of information arranged into folders. Each folder can be expanded to reveal its contents. In Figure 23, we are looking at information about Slide 6 in the demo presentation. Red dots indicate objects that need to be examined more closely because they may contain hidden data.

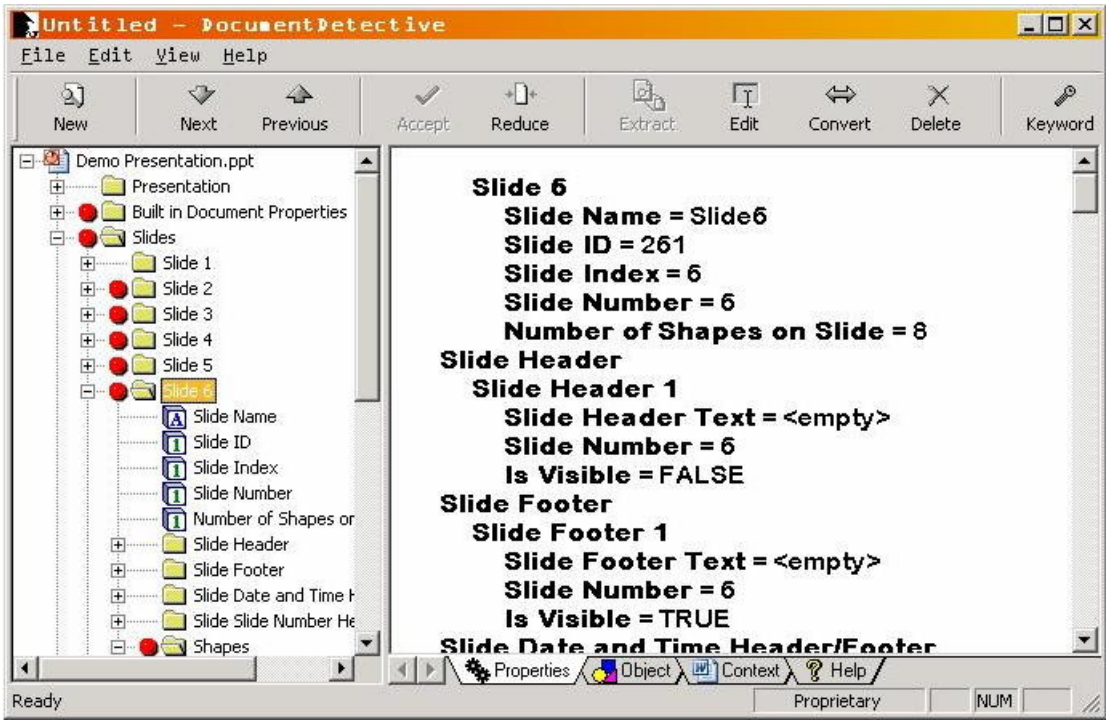


Figure 23, *Document Detective*'s document browser

The embedded objects on a slide are called Shapes. When Shape 6 on Slide 6 is selected, the properties for that shape are displayed in the Properties View as shown in **Figure 24**. This is the primary view for ***Document Detective***. All objects will have a Properties View. If the item has been marked with a Red Warning dot, the reason will be displayed in the Properties View. In this example, Shape 6 has been cropped, and the cropped area can still be recovered. ***Document Detective*** also analyzes the Notes Page for each Slide and all of the Master Slide collections. This includes the Master Slide, the Title Master, the Notes Master, and the Handout Master, which are often overlooked during an electronic document review.

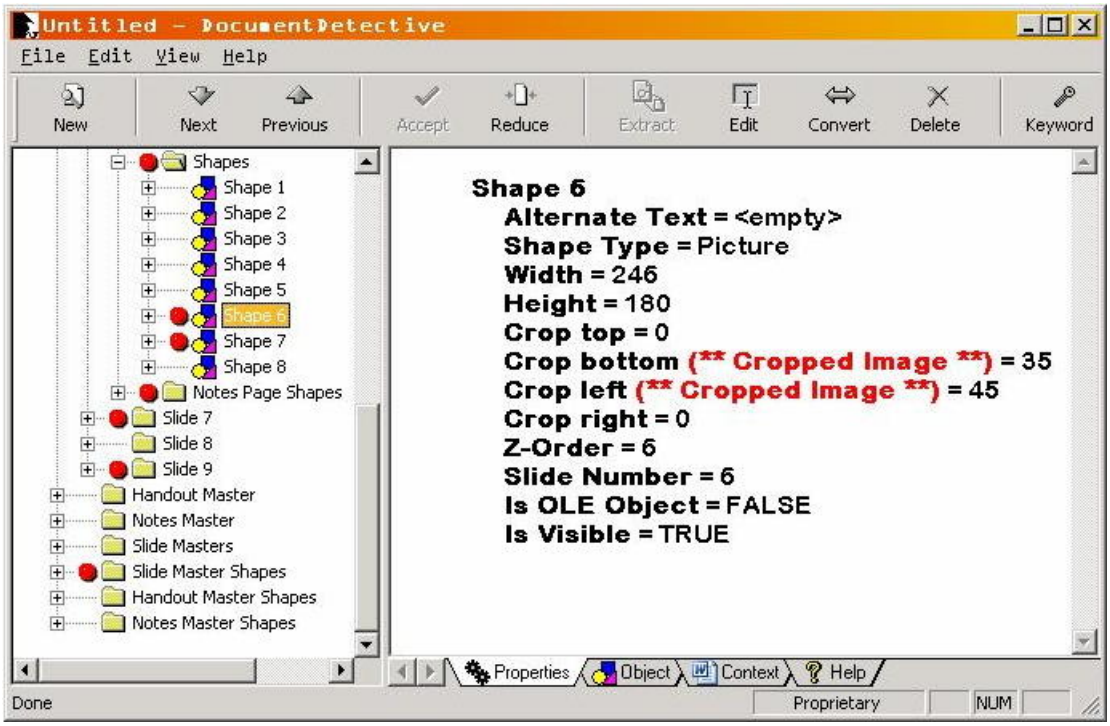


Figure 24, *Document Detective* Properties View

Many users have problems locating an object from its properties, so ***Document Detective*** also includes an Object View that displays a picture of the object as shown in **Figure 25**. Now the user should have no problem locating this object on Slide 6. The image can be resized if necessary, and scroll bars will appear as needed to pan around the image.

3.2 Word Reviews

Word Documents include a Paragraphs collection that contains all of the paragraphs in the document. The full text for each paragraph is displayed as shown in **Figure 26**. If a keyword is found, the keyword will be highlighted and the paragraph will be marked with a Red Warning dot. Having the full text available can be important when determining if the keyword constitutes a security issue that needs to be corrected.

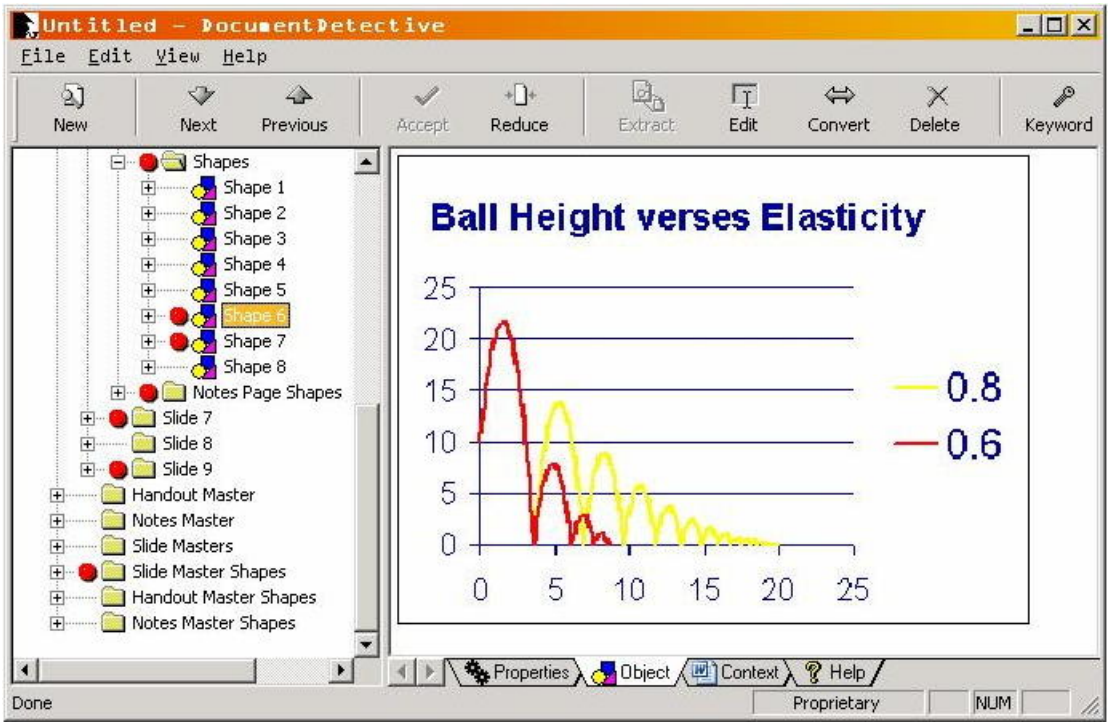


Figure 25, *Document Detective* Object View

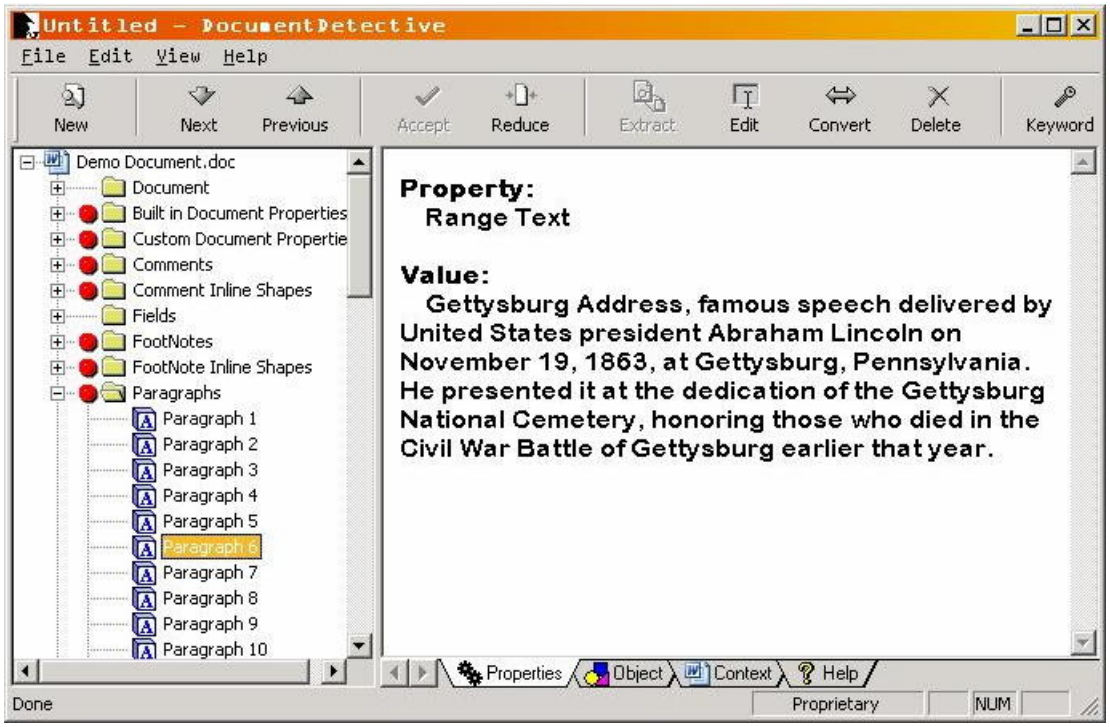


Figure 26, Word paragraph properties in a *Document Detective*

Word Documents have an additional type of embedded object called an Inline Shape. These objects are embedded directly into the text layer of the document and have different properties than a Shape. In **Figure 27**, Inline Shape 1 has been identified as a potential problem because it is an embedded OLE object. A special icon is used to mark OLE objects. In this case, the object is an embedded Excel workbook. The Extract button on the toolbar extracts and analyzes this object in a new instance of ***Document Detective***. This is a powerful feature and can be used to analyze nested OLE documents to any level, as long as it is a recognized file type.

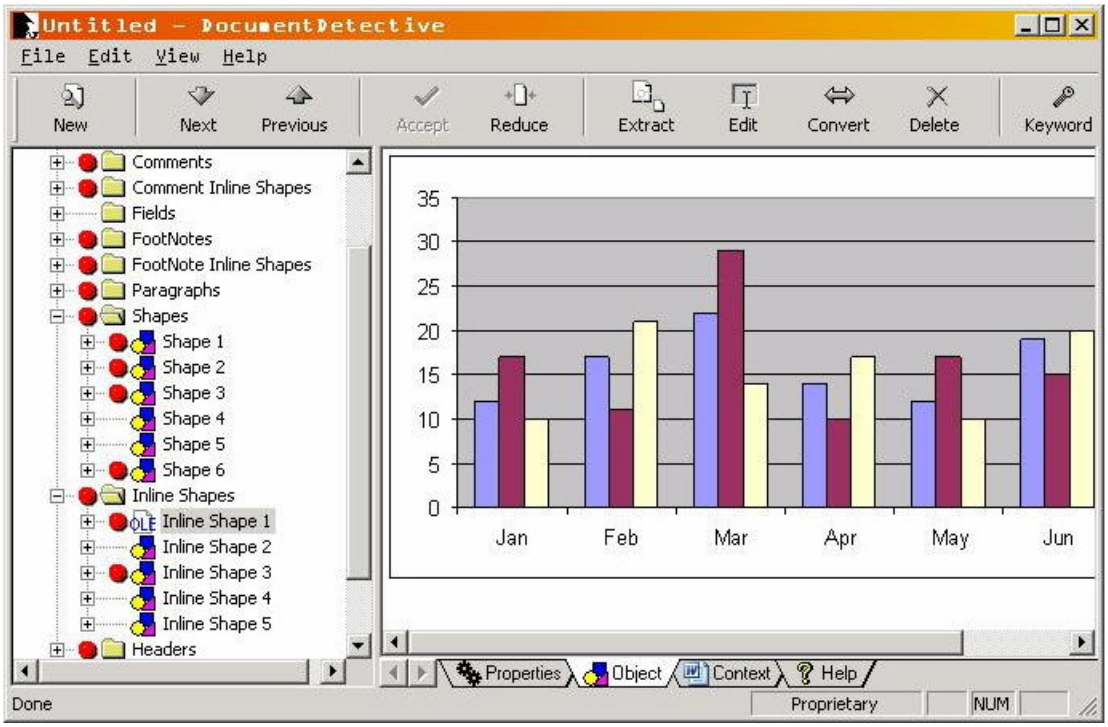


Figure 27, Word Inline Shape in a *Document Detective*

3.3 Excel Reviews

Excel workbooks are organized into worksheets and chartsheets. In **Figure 28**, ***Document Detective*** found multiple keywords on the only Chartsheet in this workbook. Every used cell in every worksheet is checked. ***Document Detective*** looks for hidden rows, columns, cells and workbooks. Cell 5 in **Figure 29** has been flagged with a Red Warning dot because it contains a keyword, and because it is in a hidden row. Worksheets and Chartsheets can contain embedded objects. Embedded objects are shown in the Shapes folders.

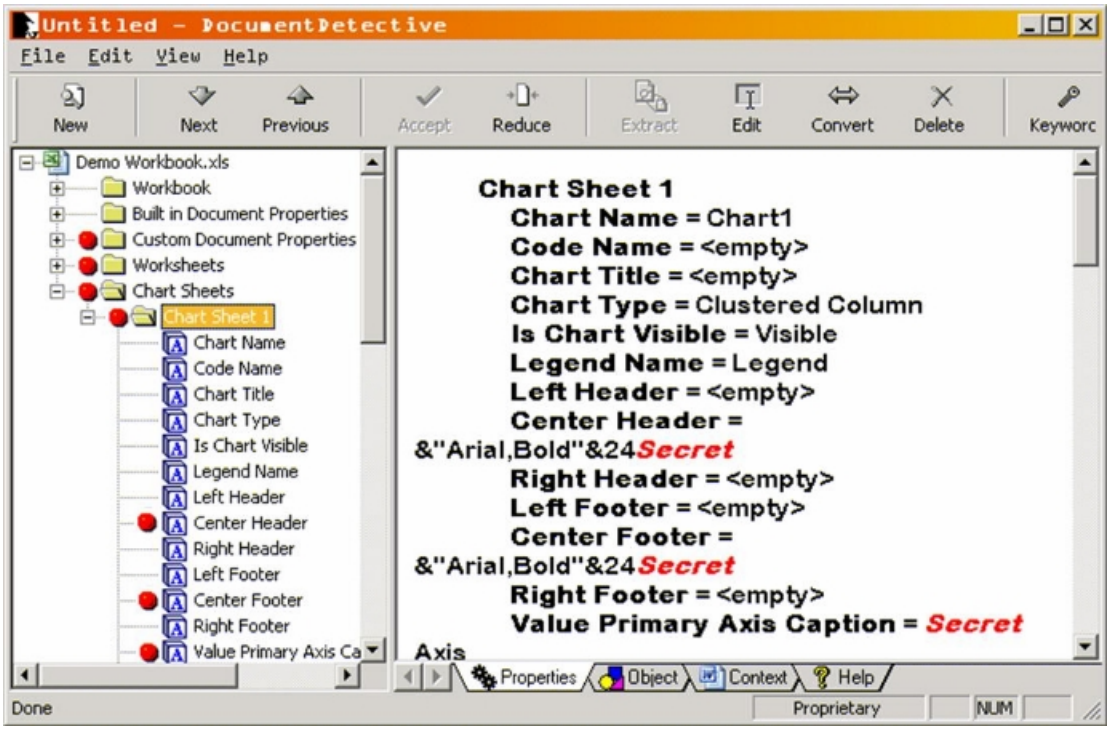


Figure 28, Excel workbook as seen in *Document Detective*

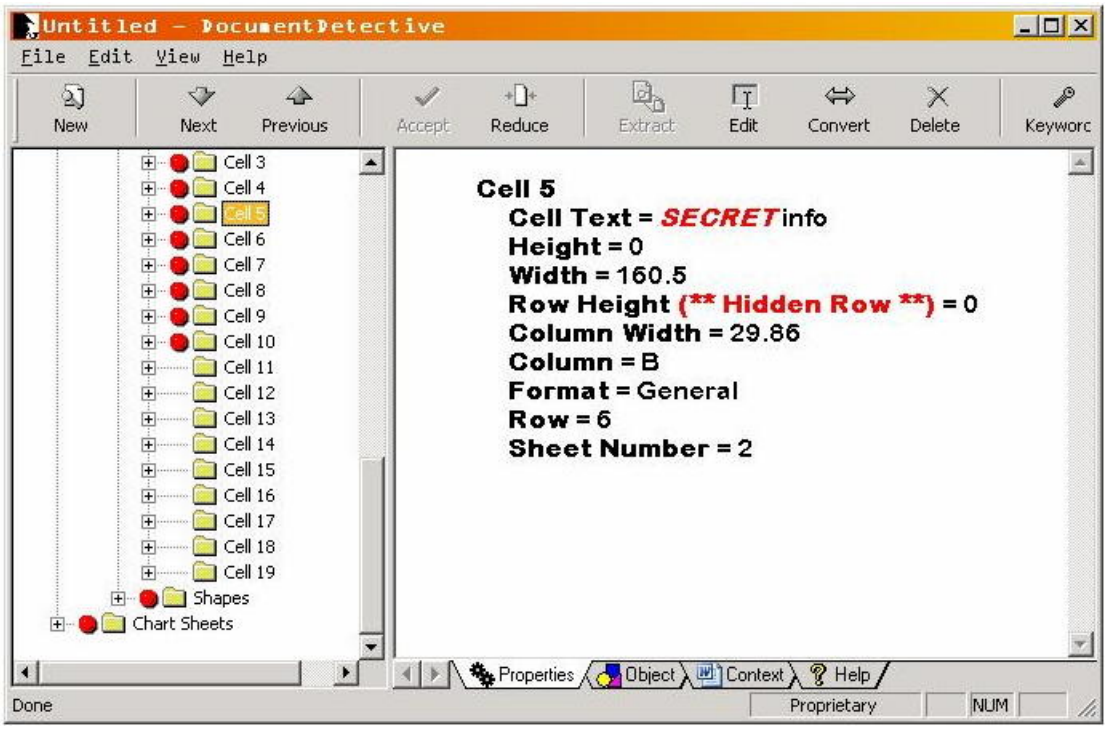


Figure 29, Excel worksheet cell properties display in *Document Detective*

3.4 Sanitization Tools

Document Detective has a built-in editor that allows some changes to be made without returning to the native application. ***Document Detective*** also adds a new toolbar to Word, PowerPoint, and Excel that helps clean up documents and reduces the file size. The most important of these new features is an advanced *Go To* button. Using the information from ***Document Detective*** with this button allows you to find problems easily. In **Figure 30**, we have entered Shape 3 into the dialog. When we click on the *Go To* button, Shape 3 is selected and brought into view as shown in **Figure 31**. This image is not visible in most views of this document. ***Document Detective*** automatically selects the appropriate view to make the object visible. Other features on the toolbar include compression, flatten, and filter options. The Compress button reduces the resolution of images, removes cropped areas of pictures, and coverts OLE objects into safer images. The Compress button allows the user to select the new image type. The Auto Compress button performs the same function as the Compress button, but it automatically selects the image type. The Flatten button uses the Auto Compress function on every image in the active document and filters the document to remove the Meta data, macros, fragments and Tracked Changes. This can result in a significant reduction in file size. The Security Scan button sends this document to ***Document Detective***.

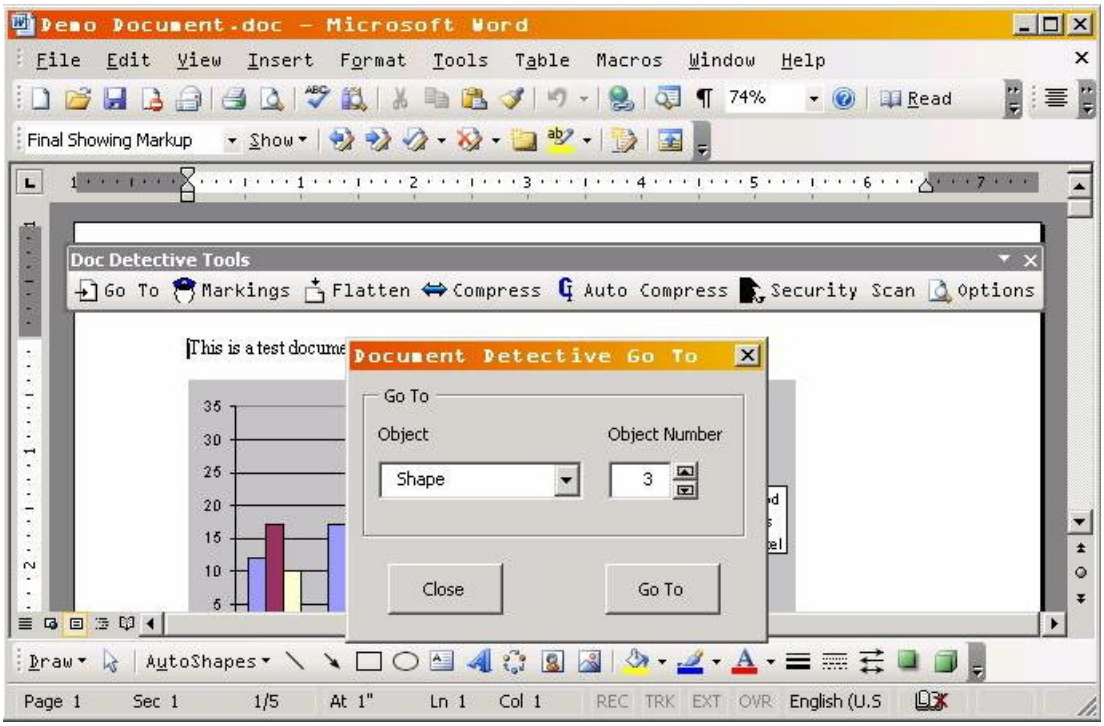


Figure 30, *Document Detective*'s Advanced Go To dialog

3.5 Adobe PDF Capabilities

Document Detective Version 1.1 now has an Adobe Portable Document Format (PDF) review capability. This Phase I PDF capability shows the text of a PDF document arranged into pages. This review tool would have prevented the inadvertent disclosure of classified information when the Special Report on the shooting of the Italian journalist in Iraq was

released. ***Document Detective*** exposed the text from the redacted paragraphs, which were still clearly marked SECRET NOFORN.

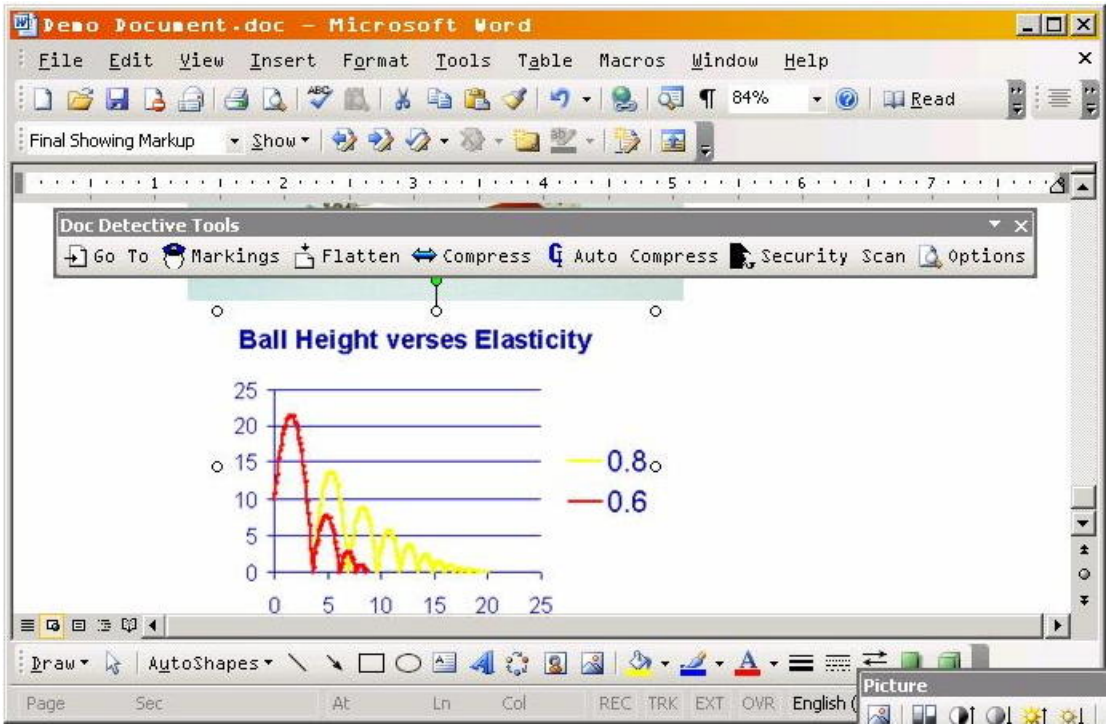


Figure 31, Hidden shape in a Word document

The text for each visible page in the PDF document is displayed in a specific Page folder under the Pages folder as shown in **Figure 32**. Folders can be expanded or collapsed for the reviewer's convenience. All text is checked by the ***Document Detective*** regular expression keyword scanner, and keywords are highlighted for the reviewer's consideration. The ***Document Detective*** PDF capability does not include an editor. If the user finds problems in their PDF document, they will need to return to the original application to make corrections.

The Phase I PDF review tool also identifies the images that appear on the page, but can not yet display those images. Images appearing on a page are found in the Images folder under the specific Page folder as shown in **Figure 33**. The height and width of the image as it is stored in the document is shown along with the encoding or compression algorithm. Images may be displayed differently because of cropping and scaling. ***Document Detective*** cannot display the images yet, but this allows the user to compare the number and size of images seen on the page with what is really stored in the document. For example, if the user sees two images on the page in Adobe Acrobat, but ***Document Detective*** reports five, there may be some hidden images on that page. Just like Microsoft Office, Adobe PDF documents can have objects overlapping other objects.

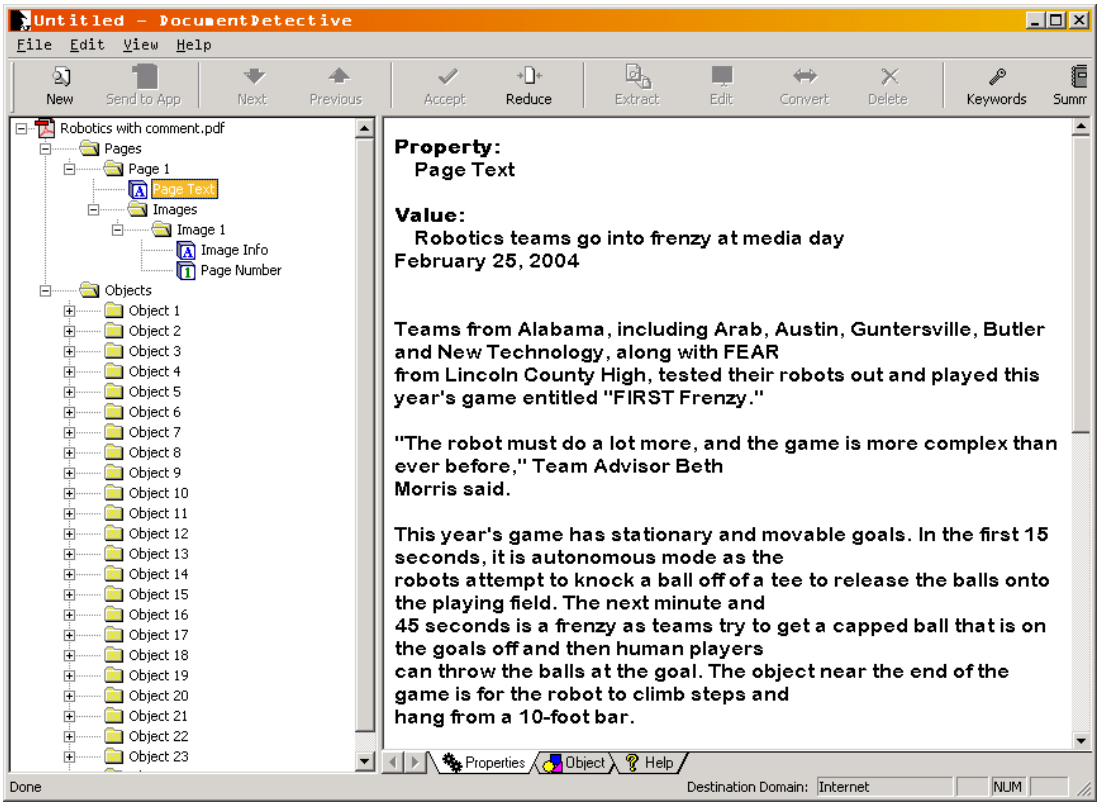


Figure 32, Text from page 1 of a PDF document in *Document Detective*

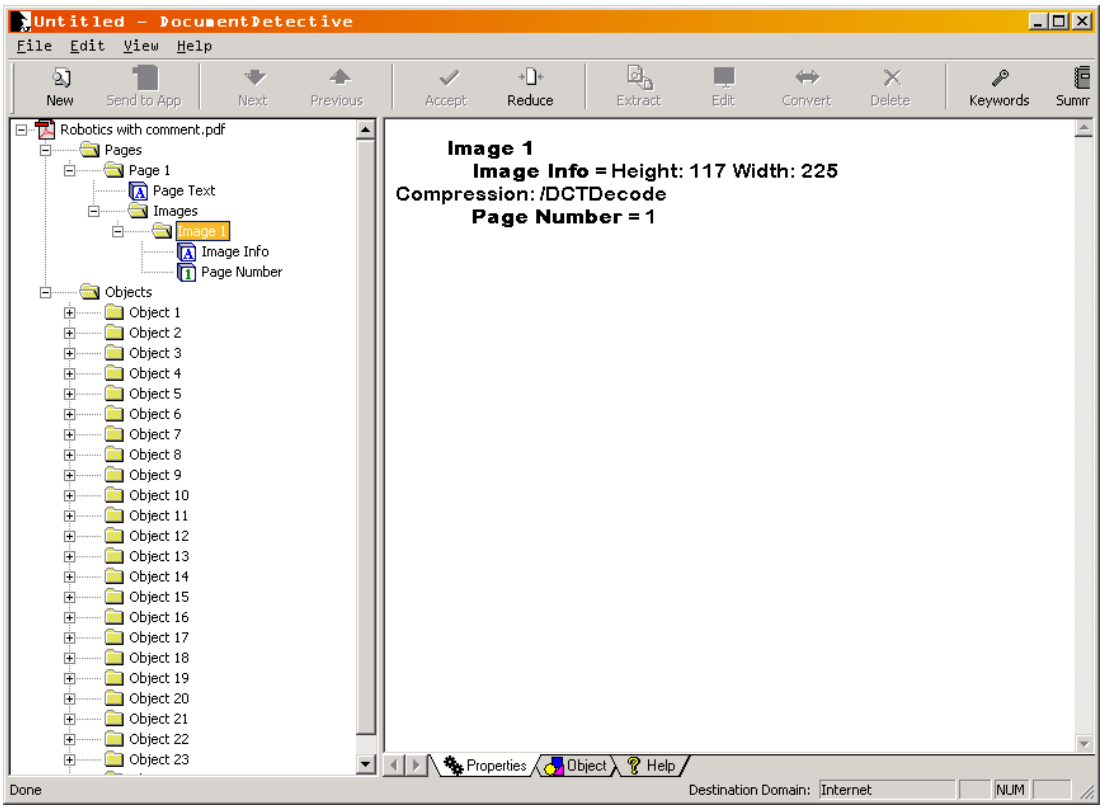


Figure 33, Image identification in *Document Detective*

PDF is an extremely complex format, and the SRS Phase I capability does not yet handle all of the objects that can be contained in a PDF document, but it does expose and organize all of the objects for consideration. ***Document Detective*** extracts all of the objects from a PDF document and then parses the objects needed for the individual page folders. The Objects folder contains all of the objects that were not used to create the individual page views in the Pages folder. A lot of this information is formatting data, font descriptors, and information that is not of interest to the reviewer, but some of the information needs to be reviewed. In **Figure 34**, we see an Annotation (similar to a Microsoft Comment). The text of the annotation is clearly visible in the object data dump. Visually scanning the Objects collection in ***Document Detective*** will alert the user to problems that could compromise sensitive information.

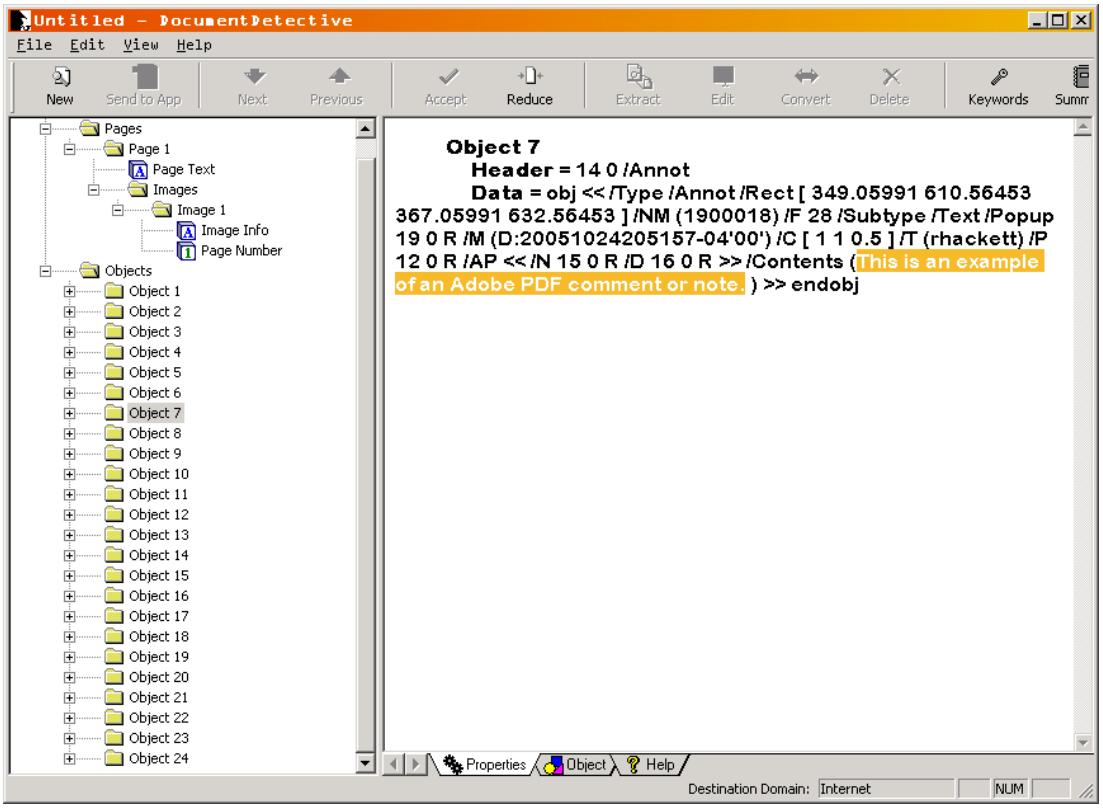


Figure 34, The PDF Objects collection in *Document Detective*

The presence of more than one Metadata object indicates the PDF document has been re-saved and contains fragments. Fragments are deleted portions of the document that may still be recoverable. ***Document Detective*** does not parse the Metadata object, but the Metadata object contains readable text as shown in **Figure 35**. The ***Document Detective*** regular expression keyword scanner does scan the unused objects, so keywords in clear text will be found.

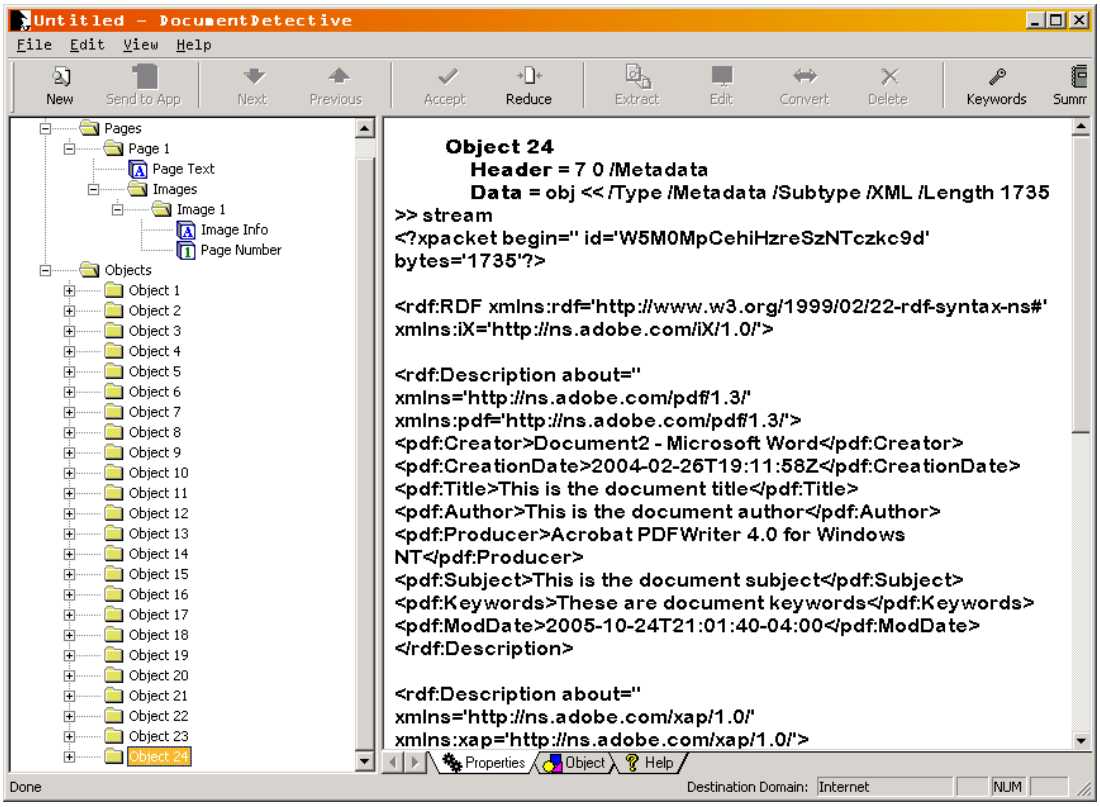


Figure 35, PDF Meta data in *Document Detective*

3.6 General Information

Document Detective was designed to support document transfers across multiple domains, such as SIPRNET, NIPRNET and the Internet. The keyword list required depends on both the initial domain and the target domain, so ***Document Detective*** allows users to build multiple keyword lists as shown in **Figure 36**.

Document Detective currently works on the following file types:

- Microsoft Word documents and templates
- Microsoft Excel workbooks and templates
- Microsoft PowerPoint presentations, shows and templates
- Rich Text Files (Microsoft Word 2.0)
- Hyper Text Markup Language (HTML) *Review Only*
- Web Archives (MHT & MHTMl)
- Extensible Markup Language (XML) *Review Only*
- Adobe Portable Document Format (PDF) *Review Only*
- Text files *Review Only*

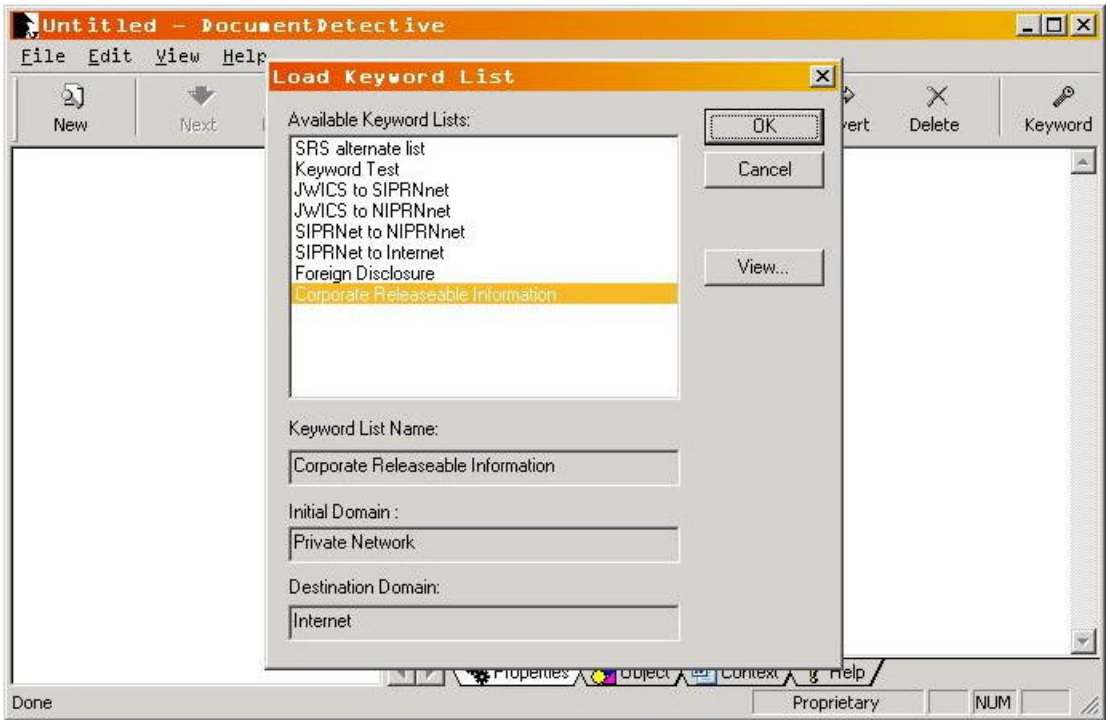


Figure 36, *Document Detective*'s keyword selection dialog

Some of the specific capabilities of ***Document Detective*** are as follows:

- Facilitates 100% reliable human review
- Uses Regular Expression keyword scanner
- Employs File Explorer type interface to browse document contents
- Extracts and examines embedded objects
- Deletes embedded objects or converts them into safer object types
- Identifies structures that are potentially dangerous to security
- Terminates Reviewing (tracked Changes) and removes Reviewing data
- Removes document fragments
- Removes Macros (Word & Excel)
- Removes Metadata
- Includes plug-ins to integrate with Microsoft Office

In addition to our document scanner and viewer, SRS has also included a robust, software controlled system for passing an electronic document from the Originator to a Reviewer to a Releaser (Security Officer) to a folder ready for transfer to another security domain. This process provides user authentication and an audit trail. It was designed around processes that have been used in the security community for many years. This system is flexible and configurable, so it can adapt to any organization's process. This feature is called the Electronic Document Review System (EDRS).

Some of the specific attributes of EDRS are:

- Implements rigorous, systematic, two-man review process
- Requires authentication for the Originator, Reviewer and Releaser
- Checks document integrity at each step of the process
- Documents the review process and generates a record certificate
- Requires the Originator and Reviewer to acknowledge all warnings
- Sorts approved documents by target domain
- Leverages Microsoft Windows NT file system (NTFS) security
- Sends progress notifications by standard email

Finally, ***Document Detective*** also includes a capability to batch process large volumes of files. The same basic engine is used, but the Graphical User Interface (GUI) is suppressed. Summary files are generated for each file processed, and a results file is built with links to the summary files.

Document Detective was designed to work with the majority of Microsoft Word, PowerPoint, and Excel documents, but because of many different configurations and features built into Microsoft software, SRS cannot guarantee that the Software will process all Microsoft Word, PowerPoint, and Excel documents. Version 1 indicates that this is the first commercial release of ***Document Detective***, so it will not have the same level of maturity as other established software.

There is inherent risk of compromising information when sharing or transferring electronic documents. While ***Document Detective*** is intended to reduce such risks, it does not completely eliminate the risk of compromising information. ***Document Detective*** must be installed and used correctly to maximize benefits.

4.0 Further Reading

The information presented here is to allow the reader to further research the issue of hidden data in electronic document and does not constitute an endorsement of the procedures and products mentioned in these publications. All of the procedures and products mentioned by the publications in this list are inadequate for reviewing electronic documents that could contain National Security Information, and they should also be considered inadequate for reviewing corporate, professional and personal electronic documents as well.

1. McCarthy, Michael J. "Beware, 'Invisible Ink' Inside Computer Files May Reveal Your Secrets." *Wall Street Journal* (Eastern Ed), Vol. 236, Issue 78 (October 20, 2000). p. A1.
2. Lemos, Robert, "Melissa Creator May Be Uncovered", *ZDNet News*, March 29, 1999.
3. Burney, Brett, "Software Shootout: Disappearing Data", *Law Office Computing*, June/July 2003.
4. "Workshare Metawall White Paper", Workshare Technology, June 2003. http://www.workshare.com/collateral/product/whitepapers/US_WhitePaper_Metawall.pdf
5. "Case Study: Workshare Metawall and Gibbons, Del Deo, Dolan, Griffinger & Vecchione", June 2003. http://www.workshare.com/collateral/case%20studies/US_Case%20Study_GDD.pdf
6. "iScrub - Premier Metadata Removal Utility For Microsoft Office", Esquire Innovations, Inc., June 2003. http://www.esqinc.com/products_iscrub.asp

7. Rindsberg, Steve, "The PowerPoint FAQ", 22 July 2003. http://www.rdpslides.com/pptfaq/FAQ00062.htm
8. "Coping with Metadata Issues", Microsystems, June 2000. http://microsystems.com/Shares_Well.htm
9. Knight, Will, "Online Document Search Reveals Secrets", NewScientist.com, 15 August 2003. http://www.newscientist.com/news/news.jsp?id=ns99994057
10. "OFF97: How to Minimize Metadata in Microsoft Office 97 Documents", Tech Article Q223396, Microsoft Corporation, November 13, 2000.
11. "WD2000: How to Minimize Metadata in Microsoft Word Documents", Tech Article Q237361, Microsoft Corporation, December 29, 2000.
12. "Trusted Download", Defense Security Service, 15 August 2002. http://www.dss.mil/infoas/trusted_download_072602.doc
13. Hackett, Ronald D., "The Desktop Publishing Threat to Information Security", Redstone Arsenal & NASA Marshall Space Flight Center IT Security & Assurance Conference and Exposition, 12 July 2001. https://www.technologyforums.com/redstone-nasa/IT%20Expo%20Brief%20for%20Release.pps
14. Turner, Adam, "No Back Doors for CIA in Our Code: Microsoft", *The Age*, 15 August 2003. http://www.theage.com.au/articles/2003/08/15/1060871752574.html
15. Brown, Russell, "The Word on Iraq," *New Zealand Listener*, Vol. 189, No. 3297, 19-25 July 2003. http://www.listener.co.nz/default,376.sm
16. Smith, Richard M., "Microsoft Word Bytes Tony Blair in the Butt", 30 June 2003. http://www.computerbytesman.com/privacy/blair.htm
17. Ward, Mark, "Tools Reveal Secret Life of Documents", *BBC News*, 3 July 2003. http://news.bbc.co.uk/2/hi/technology/3037760.stm
18. Ward, Mark, "The Hidden Dangers of Documents", *BBC News*, 18 August 2003. http://news.bbc.co.uk/2/hi/technology/3154479.stm
19. Leonhard, Woody, "Office XP/2003 Hidden Data Removal Tool," *Woody's Office Watch*, Vol 9, No 1, Article 2, 9 January 2004. http://www.woodyswatch.com/office/archtemplate.asp?v9-n01
20. Hackett, Ronald, "Review of Microsoft's New 'Remove Hidden Data' Tool", SRS Technologies, 13 January 2004. http://www.stg.srs.com/eds/RemoveHDToolReview.pdf
21. “Danish Prime Minister Gets Bitten by Word ,” The Sydney Morning Herald (smh.com.au), January 13, 2004. http://www.smh.com.au/articles/2004/01/13/1073877800625.html
22. Leonhard, Woody, "MS Hidden Data Removal Tool not Ready for Prime Time," Woody's Office Watch, Vol. 9, No. 2, Article 1, 20 January 2004. http://www.woodyswatch.com/office/archtemplate.asp?v9-n02
23. “Poor Document Collaboration Adding Serious Risk, Cost to Business,” Workshare, 15 Dec 2003. http://www.workshare.net/news/ne_pressreleases_single.asp?pressID=81
24. Johnson, Nick, “Alcatel [Expletive Deleted] Up Bigtime,” Morons in the News, Apr. 14, 2001. http://web.morons.org/article.jsp?sectionid=1&id=188
25. Shankland, Stephen and Ard, Scott, “Document shows SCO prepped lawsuit against BofA,” CNET News, March 4, 2004. http://news.com.com/2100-7344_3-5170073.html
26. Jardin, Xeni, "P2P in the Legal Crosshairs," Wired News, 15 March 2004. Note: Skip to paragraph 5 to see why we listed this article. http://www.wired.com/news/digiwood/0,1412,62665,00.html?tw=newsletter_topstories_html
27. Zalewski, Michael "Strike that Out, Sam", 29 March 2004. http://lcamtuf.coredump.cx/strikeout/
28. Wildstrom, Stephen H., "Don't Let Word Give Away Your Secrets," Business Week, pg 26, 19 April 2004. http://www.businessweek.com:/print/magazine/content/04_16/b3879047.htm

29. Kernan, Deborah, "Hidden Data in Electronic Documents," SANS Institute, 5 July 2004. http://www.sans.org/rr/papers/14/1455.pdf

30. Hayes, Simon, "Canberra crackdown on Office leaks," Australian IT News, 20 May 2004. http://australianit.news.com.au/articles/0,7204,9608307%5e15319%5e%5enbv%5e15306,00.html

31. Metadata Risks (http://www.metadatarisk.org), sponsored by Workshare.

32. Kaplan, Ari, "A New Generation of Redacting Tools," The National Law Journal, 14 Nov 2002. Note: the first three paragraphs are pertinent. http://www.law.com/jsp/printerfriendly.jsp?c=LawArticle&t=PrinterFriendlyArticle&cid=1036630382605

33. Hackett, Ronald D., "The Desktop Publishing Threat to Information Security," Redstone Arsenal & NASA Marshall Space Flight Center IT Security & Assurance Conference and Exposition, 12 July 2001. https://www.technologyforums.com/redstone-nasa/IT%20Expo%20Brief%20for%20Release.pps

34. Hackett, Ronald D., "Electronic Document Security: The Desktop Publishing Threat and Mitigation Strategies," Redstone Arsenal & NASA Marshall IT Security & Assurance Conference and Exposition, 14 May 2003.

35. Hackett, Ronald D., "The Desktop Publishing Threat to Information Security," Air Intelligence Agency National Information Systems Conference, San Antonio, TX, 8 July 2003.

5.0 Acknowledgements

Initial development of the ***Document Detective*** prototype software was supported by the National Science Foundation, Small Business Innovative Research grant number 0232955.

6.0 About the Author

Ronald D. Hackett is a Program Manager and Electrical Engineer for ManTech SRS Technologies in Huntsville, Alabama where he has been working since he retired from the United States Air Force with over twenty years of service in systems engineering, program management, and Intelligence. Ron earned his Bachelor of Science degree in Engineering specializing in Digital Electronics and Computers at the University of Central Florida, and he holds a Master of Science degree in Electrical Engineering from the University of Dayton where he studied Antenna Theory and Electromagnetics. He has also studied business at Webster University in St. Louis, Missouri. Mr. Hackett is a registered Professional Engineer (PE) in the states of Ohio and Alabama, a Certified Cryptologic Engineer by the National Security Agency, and is a Senior Member of the Institute of Electrical and Electronic Engineers (IEEE) where he serves as a Director for the Joint Engineering Council of Alabama (JECA).

Mr. Hackett's last active duty assignment was with the Missile and Space Intelligence Center (MSIC), Redstone Arsenal, Alabama where he was responsible for reviewing electronic documents being transferred between Top Secret, Secret, and Unclassified security domains for hidden information that could compromise classified information. He quickly developed an extensive understanding and expertise in this area that is unparalleled in Government or industry. Then Major Hackett prevented numerous compromises of National Security Information through his diligence and perseverance. Since retiring from the US Air Force, he has been working on improved methods of detecting, analyzing, and controlling the hidden data in electronic documents. Based on his experience with MSIC and his continuing research with SRS, Mr. Hackett is more concerned than ever about the potential for compromising sensitive and classified information when sharing electronic documents.