

Microsoft® Office System XML File Formats

Introduction

This document provides information regarding Microsoft Office System XML file formats. Unlike HTML, XML is a meta-language enabling companies to define their own file formats through the creation of custom-defined XML schemas. The Microsoft Office System supports the creation and manipulation of an unlimited set of XML file formats as needed by customers to define the content of their business documents. The Microsoft Office System also supports two fully documented application-specific document schemas (Word ML, introduced in Office 2003 and Spreadsheet ML, introduced in Office XP) that primarily function to store information about document display and Word and Excel specific functionality pertaining to specific Microsoft integrated innovations.

Abstract

With many companies and industries having seen great benefit from the use of XML, primarily for exchanging data between back-end servers, the introduction of XML-enabled *desktop* applications with support for customer-defined schemas carries the potential for more cross-platform integration for even greater data access and collaboration benefits. The growing industry trend to support customer-defined schema is enabling information workers to create and interact with documents that contain regions of meaning, in the same way that information in a structured or relational database has meaning. With such support, XML brings the power of traditional data management to bear on documents, facilitating reuse, indexing, search, storage, aggregation, and other practices more often associated with management of relational databases. The Microsoft Office System's XML support for customer-defined schemas in the 2003 versions of the Office applications puts XML in the hands of information workers and enables businesses to reap the full benefit and promise of this revolutionary technology.

1. The Value of XML

Today, XML is a widely accepted industry standard that enables exchange of data between disparate systems. The use of XML and the advent of the XML Web services architecture is set to revolutionize the way many companies—or, in some cases, entire industries—transact business on the Web. However, XML represents more than a simple data or document exchange protocol; its founders originally envisioned XML as a way to capture more of the *meaning* locked within business documents by defining the structure and context of the information these documents contain instead of only describing the way those documents are displayed.

Although there are well-established methods for storing and managing data types (for example, numerical data in databases), a significant portion of the information created in the business environment is not captured in any meaningful way and thus can't really be accessed or reused. Workers everywhere generate reports, e-mail messages, documents and spreadsheets that contain vital, valuable information. But if they need to reuse this information, these same workers may also spend significant time searching for the appropriate files and subsequently spend time and effort to re-key, cut, paste, or otherwise import the relevant information into another document. The way these documents are created and handled tends to limit the extent or ease with which the information can be used outside the original document.

While data capture and validation is a well established methodology for traditional data management, the technology to similarly gather and manage the information contained in text-based reports and other common business documents has not been available. This was the problem that XML's creation solves, through enabling the use of custom-defined schemas. XML enables businesses to capture all manner of business information in a way that maximizes its value. By facilitating reuse, indexing, search, storage, aggregation, and other practices more often associated with management of relational databases, XML brings the power of traditional data management to bear on documents.

XML enhances interoperability across heterogeneous and cross-platform systems at the new and fundamental data level for documents, allowing data to be restructured, aggregated and presented in new and different ways rather than simply allowing a second platform to display the same data in the same order.

2. How does XML enable the creation of custom document file formats through custom-defined schemas?

XML is a *markup language* that is used to identify structure within a document. The XML standard is published and maintained by W3C, the consortium that maintains many of the standards for the World Wide Web. Microsoft has designed its Office applications to adhere to the W3C's published standard.

Like other markup languages, XML uses *tags* to define specific elements within a document. XML tags define the document's structural elements and the meaning of those elements. Unlike HTML tags, which specify how a document looks, or is formatted, XML can be used to define the document structure and content—not just the look and feel. By doing so, XML separates a document's content from its presentation, thereby enabling developers to access and manage this content in many meaningful ways.

The tags that can be used for a particular document type or information type are contained in what are called *XML schemas*, which define the set of tags and the rules for applying them. Schemas, thus, define the structure and type of data that each data element in a document can contain and schemas can be created by a number of entities such as a user, a company, or an industry for example.

With that understanding in mind, XML schemas can be created to define and qualify content for virtually any application. For example the information that can be found in documents about finance, insurance, health care, automobile manufacturing, government regulation, insurance claim processing or visa applications can be very different (or identical) for each type of document. In order to best describe the inherent differences, different XML schemas are needed to describe the type of information in each document. While the underlying structure of a document enriched with schemas is XML, defining the type of information in documents is in essence creating a file format for each different type of document: Financial tags such as <Amount> or <ROI> will be used in a file format for financial documents, and tags such as <DateOfTreatment> or <PatientRecord> for healthcare documents.

With the use of custom-defined XML schemas, needed information can easily be properly extracted from any document at any time and interchanged with any organization or other application using database-related techniques.

In spite of this potential, XML has yet to be fully exploited through desktop applications. With the XML support for customer-defined schemas now introduced in the 2003 versions of the Office applications, customers can begin to reap the full benefit and promise of this revolutionary technology through *better capture and reuse of information, more easily connecting to data and intelligent applications*.

Capture and Reuse of Information

XML-based documents enable organizations to capture more of the intellectual property that is created on an ongoing basis. Customer-defined schemas are analogous to the columns and tables in a database; thus, documents of all kinds become a source of information as rich as any other operational data store. Once captured, this information becomes a very valuable corporate asset. By defining their own XML schema, organizations gain the ability to decide exactly what data to capture and now this data is structured. With documents functioning at this level of "storage," companies have the capability to aggregate, parse, search, manage, and reuse documents, document fragments and domain knowledge in the same way they do their other business data.

For users, the ability to search for specific information and to aggregate information from numerous sources eliminates many of the time-consuming, error-prone tasks associated with document creation and update; for example, opening and closing files to find information; cutting and pasting information between documents; and searching for labels to combine data in like fields.

Connecting Users to Data

XML is widely regarded as a standard for data exchange and for exposing the information contained in databases or back-end systems. Built on open, industry standards, today's XML web services provide a universal way to connect users to data in order to allow

communication between business systems and data sources, or between systems that are written in different languages on different platforms. By providing XML-enabled applications on the desktop, companies can take advantage of a Web services infrastructure to empower employees and enable them to connect directly to enterprise systems and data sources.

Using Office technologies, companies can take full advantage of XML data and XML Web services by accessing the information directly and then dynamically surfacing the right information, in the right form, to where it's needed in the spreadsheet or word processor for analysis, formatting, publishing or other type of processing. The result is a flexible, richer, more integrated desktop environment. Throughout, the information retains its meaning from customer-defined XML tags (healthcare tags such as <PatientRecord> retain their meaning in the word processor or in the spreadsheet, even if the <PatientRecord> data is presented in a formatted bold paragraph in the word processor or in a table grid in the spreadsheet). Without these, the information would become a random group of bytes that could not be subsequently reused in an intelligent or automated fashion.

Intelligent Applications

XML offers exceptional potential for automating virtually any task that involves working with documents. Creating documents such as reports, spreadsheets, and forms with an attendant XML schema—even if that schema is hidden to the users—enables developers to build interoperable solutions that recognize the structure and meaning of the content within those documents and respond intelligently to the user. Information from customer-defined schemas can also be used to validate information or data as it is entered, avoiding errors and aiding in data cleansing and standardization.

The ability for companies to define their own schemas allows them to identify the unique regions of meaning within their documents and to create solutions that correlate these structures to their own business processes. Because the actual content is separate from the presentation of the content, these solutions can be tailored to display the same information in many different ways, as appropriate for a particular task, user, or process.

Moreover, the ability to identify sections of a document structurally—or to recognize specific content within a section—allows developers to create applications that respond intelligently to user input, offering context-sensitive actions and guidance, suggesting required content, or providing supporting data or links to related information.

Because the client software understands the content in the document, through the custom-defined XML tags, intelligent applications present endless possibilities for helping users interact with documents. The advent of such solutions will revolutionize the way users create and work with documents. Intelligent applications will guide and facilitate the creation of documents, reducing the time spent on traditionally manual tasks.

3. Microsoft Word, Excel and Access support for custom-defined XML schema

XML support in Microsoft Word 2003 enables authoring and saving of rich content in custom file formats based on customer-defined XML schemas, enabling the repurposing of document content across devices, platforms and processes. Support for customer-defined XML schemas enables users to preserve or extract from the document selectively the data or structural elements of interest to a particular application. In either case, users can create documents containing information marked by XML tags belonging to the custom-defined schema in a completely intuitive fashion; users need not learn or understand the concepts behind XML to realize the full benefit. Organizations, on the other hand, greatly benefit from the aforementioned advantages of using a custom-defined XML schema to insure that the information the user enters is of a high quality, has contextual meaning for the business process at hand and can be easily reused.

Customers who use Microsoft Excel 2003 for importing and analyzing business data will benefit from the enhanced XML capabilities introduced in this version. Like Word 2003, Excel 2003 can read custom XML files as defined by customer-defined schemas. This enables Excel 2003 to act as a smart client for XML Web services and a host for smart document solutions that require analytical and calculation capabilities rather than rich text formatting.

Additional Excel XML capabilities include a visual tool for ease of mapping between the spreadsheet grid and customer-defined XML schema. This enables developers or power users to more easily import or export data in Excel to or from enterprise data stores or Web services.

Microsoft Access 2003 enables Office users to import and extract XML data from database tables using custom-defined schemas and XML transforms¹. Access 2003 also enables the creation of a custom-defined XML schema derived from the database schema. These capabilities facilitate the integration of Access data with related business processes or documents and allow users to control exactly how the data is represented in XML.

Other Uses of XML in Microsoft Office System applications

New with the Microsoft Office System, InfoPath 2003 uses a forms metaphor to capture information according to a customer-defined XML schema. InfoPath enables customers to gather and reuse information with predefined structure (pre-tagging) and as part of a business process. InfoPath supports only XML file formats based on customer-defined schemas enabling users to interoperate with any Microsoft or non-Microsoft platform that produces or consumes XML files belonging to the customer's XSD.

FrontPage 2003 lets users quickly build high-quality, data-driven Web sites that present dynamic views of information from enterprise systems or local data stores. FrontPage supports a complete set of tools for creating and editing Web pages that connect to a variety of data sources, including XML files that follow customer-defined XML schemas, databases and XML Web services. Users control how data will be displayed in a Web page by creating XSL-T transforms using an intuitive, graphical editor. These data views include industry-standard reporting tools for sorting, grouping, filtering, and conditionally formatting data. By supporting XML files that follow customer-defined schemas, FrontPage enables users to interoperate and construct web sites using data that have been created on Microsoft or non-Microsoft platforms.

Visio 2003 drawing and diagramming software gives users the capability to integrate information from a database into a diagram. Diagrams saved as Visio XML files could incorporate XML data that follows a customer-defined schema and can later be mined to retrieve data from within the diagram. This enables developers to create rich Visio solutions for modeling business processes, or that associate data from any XML data source with specific shapes or diagram elements.

Interoperability and Heterogeneous, cross-platform data interchange

Support for custom-defined schema in Office 2003 is the fundamental enabler for data interoperability. Documents can be created in Office 2003 following the XML format defined by the customer using the W3C XSD standard. Any Microsoft or non Microsoft XML environment, client or server that support the W3C XML and XSD standards can then consume those documents. XML documents created by Microsoft or non-Microsoft systems belonging the customer-defined XSD can be read and analyzed by the Office 2003 system. The wide adoption of XML standards along with XML customer-defined schema capabilities across the Microsoft Office System open doors to many new innovative applications that can lead to better use and reuse of information.

4. Word and Excel support for Application-specific XML Document Schemas

In addition to Microsoft supporting the W3C XML standards and integrating innovations such as custom-defined schema and XML Web services, Microsoft takes XML to another dimension by offering Spreadsheet ML and Word ML to provide customers with added functionalities.

Microsoft Excel 2002 introduced Spreadsheet ML, a display-oriented XML file format that uses XML tags to store display and presentation characteristics and spreadsheet functionality. For example, this display-oriented XML schema uses a <cell> tag and a <row> tag. This file format is useful in scenarios where customers want to dynamically construct a spreadsheet file on a server without using Excel directly, which can be done using XML. However, while data can be easily accessed and retrieved, it is difficult for this spreadsheet display format to be used in output or storage scenarios since the row, cell and column information isn't descriptive enough for subsequent business use. Data is better expressed with XML tags chosen by an organization and

¹ XML transforms are software programs written for example using the W3C XSL-T standard, that enable conversion of an XML file following a specific schema to another XML file using the same or a different schema.

reflecting the content of the data (such as a <price> tag or a <Monthly-Results> tag). Early indications from the Microsoft Office 2003 beta program lead us to expect that the majority of Excel users will use the new customer-defined schema capabilities to import and analyze XML data in Excel 2003. The XML file format for Excel 2003 stores the customer-defined schema information in the same file with the other spreadsheet XML tags and standard XML techniques and tools can be used to easily extract any subset of information for reuse.

Microsoft Word 2003 introduces Word ML, a display-oriented XML file format that preserves the formatting and presentation of the Word document, including formatting, hyperlinks, paragraphs, tables and styles. Word ML also provides storage information for the entire feature set of Word 2003, including the new, advanced capabilities around smart tags, smart documents and range permissions. In the same way as with Excel, we expect Word ML to be used in a limited context to create or preserve the formatting of documents, however most customers are expected to use the support for customer-defined XML schemas along with Word ML. The Word XML file format stores the customer-defined schema information in the same file as Word ML to allow customers to easily extract the information they need while being able to easily manage the storage of one physical file.

For example, one could imagine transforming Word ML to another display-oriented schema. Other useful scenarios include server-based processes that add Word ML markup to existing text files, XML files or data for display purposes, transforming any XML document to a Word ML document, generating Word ML documents without using Word or creating a Web service that produces documents in Word ML format. All these are geared toward displaying data in a rich format in Microsoft Word 2003.

Microsoft provides full documentation of the WordML and Spreadsheet ML schema file formats. We expect Word ML and Spreadsheet ML to grow with each new version of Word and Excel and market adoption and feedback will ensure a continuous growth of the file formats to reflect the new functionalities and innovations of the new versions of the products.

5. XML and the Interoperability of Rich Authoring Tools

Unlike HTML, XML is a meta-language enabling customers to define their own data-interchange, document file formats, allowing customers to achieve data exchange interoperability in a heterogeneous environment. XML also permits software vendors to differentiate their product offerings through innovation in how data is presented or displayed even as they support data interchange interoperability through support of customer-defined schema. This promotes innovation and competition between product offerings, which is a benefit to customers.

Contrast this with what would happen if there were only one schema, which controlled both how data interchange occurs and information on how the data must be presented. Customers would be unable to define the business-specific organization and display of information that they needed, and additionally, innovation in presentation and display of data by vendors offering software products and services would be inhibited.

In such a case, some of the tags in a document refer to presentation and display functionalities (such as table editor features or a page layout editor) that have to be implemented by every tool with every detail. If a standards body picks a set of presentation and display tags that are supported by one vendor, it could disadvantage other vendors who might have presentation and display functionality that is more preferable for their customers.

An example of a problem that could be encountered in the above scenario is the following: Assume that one product supports a presentation and display feature to enable the automatic positioning of all referenced images at the end of a document. If a user of another product, which does not support this feature, tries to open a document that uses this feature, the images will be at best displayed inline in the document; at worst they would not be displayed at all. When the user edits the file, he could be changing important paragraphs that are linked to images he cannot see. The user may not even be aware the document has been changed because he edited page 1 and the images were supposed to be in page 4. To the user who originally created the document, this will seem like document corruption. The vendor of the second application or the user might be under the impression that the vendor of the application with the feature has made it "too difficult" to share files by virtue of its implementing the new feature. This situation underscores a fundamental tension between the desire for display and presentation-oriented uniformity, and the competing desire that users express for new innovative display and presentation features that provide greater value. As with many complex issues, a solution likely rests with an understanding that display-oriented

consistency should be pursued to the extent possible without sacrificing or undermining the software industry's ability to innovate and provide greater value for its customers.

Despite the challenges inherent in achieving both data interoperability and display/presentation consistency between products, there has been amazing progress over the last several years, both in the ability to achieve rich data exchange without loss of data context, as well as the ability to achieve the exchange of documents while preserving increasing levels of display and presentation consistency. Today, customers of XML and XML Web services create custom-defined schemas and then use technologies such as XSLT (XSL transforms) or products such as Microsoft BizTalk server, IBM WebSphere, WebMethods or BEA WebLogic to implement easy transformations between custom-defined schemas or display oriented schemas.

6. The Tradeoffs of Mandating a Standard XML Document Format

Some have debated the merits of establishing a "standard" document format that would be enforced by government or legislative mandates, a discussion driven in part by the legitimate desire on the part of computer users for improved interoperability between competing products. But efforts to mandate a single, vendor neutral data and display format for all documents would stifle software innovation by causing all software to allow reuse and display of data according to a "lowest common denominator." This approach would likely decrease competition over time, it is notable that many open file formats exist today but each has followed a history that balances interoperability and innovation, Open formats such as HTML, Rich Text Files (RTF) and ASCII each promoted data interoperability across competing products, but none were ever expected to achieve both data interoperability and display/presentation uniformity, nor were they typically used as the default formats for saving a document. Instead, these formats were only presented as yet another option for consumers to choose from, to facilitate interoperability and file exchange at a data level.

The history of HTML explains in part why mandating a single, vendor neutral presentation/display format proves limiting in practice. Since HTML became a standard a few years ago, there has been essentially no innovation in HTML. A comparable XML presentation/display standard would slow innovation around XML and create friction for users of documents, depriving users of the rich data mining potential that XML offers. Mandating a single standard XML document format would impact users by restricting their definitions of business-specific data and allowing them to only employ a small set of document editing features for display and presentation that are supported by all rich authoring tools. Tools vendors would not have an incentive to create innovative document-related software that could help increase organizational productivity through new display and presentation mechanisms because users either couldn't take advantage of these new capabilities, or would find it difficult to do so.

Imagine, for example, if the standard were set to support only lowest common denominator features. This would sacrifice value and richness for greater display and presentation uniformity but the lack of opportunity for vendors to add their unique value might diminish competition over time as well. On the other hand, an approach based on using custom-defined schemas enables vendors to add competitive features to their display and feature oriented XML format and enables customers to use a mapping between their custom-defined schema and each vendor full feature oriented product. Interoperability happens at the custom-defined data level.

At the other end of the spectrum, standardizing on a very rich XML file format that represents document display characteristics can lead to poor user experiences as well. The process of reading and writing a given XML file format should be fairly easy for software vendors to do, allowing most to claim "compatibility" with a file format standard for data interchange purposes. However, it would be difficult for all the authoring programs to fully support all the *display and presentation features* specified by a rich document file format that went beyond the goal of data interoperability and exchange and provide unique, innovative value to the customer - which could again reduce competition and customer choice over time. One possible outcome could be that users will have different and inadequate views of documents depending upon which authoring tool they are using. For example, users could be faced with document graphics that display at different places in a document or in different sizes, reviewers' comments or ether types of document notes that don't appear, or compound document information that doesn't get assembled correctly when pulling together sections from separate document files. Moreover, in the authoring process, users are clever about finding a program feature that enables them to achieve the desired structure or effect they need for their document; however, they cannot be expected to know which features of an authoring program might map back to approved file format characteristics.

As originally envisioned, XML schemas were intended not as the basis for a single, all-

encompassing standard for file formats. Indeed, XML's greatest promise is in enabling every organization to define custom schema that best represent the *data* that makes sense for their business. While this assumes a proliferation of schemas, the fact that each schema is based on XML allows this to occur without sacrificing interoperability.

In speaking with customers, Microsoft has learned that for most organizations, capturing the meaning of their data using custom-defined schemas is more important than having a common display-oriented document format. With this in mind, most technology companies today are focused on providing *data* interoperability. However, this focus does not mean that software such as Microsoft Office System 2003 will not provide effective mechanisms for enabling greater *display and presentation* consistency as well. For example, a standard XSLT could be created to always format the data from one agency to have a common look for published materials, and in fact a different XSLT could be used to format the data differently depending on the display medium (paper, computer application screen, Web browser, PDA).

In the public sector, Microsoft and other industry partners encourage government agencies and parliaments to specify how their documents should appear by defining as many XSLT style sheets as are needed, using any display characteristics available on the market, and more importantly maintaining them, changing them and making them evolve at will. This approach of enabling public sector agencies to define exactly how public documents will be displayed is far better than requiring all customers to generate documents in a static display-oriented schema that cannot evolve except by the agreement of a standards committee.

7. The Value of Promoting Multiple Technical Approaches for XML File Formats

Another example of how a single format for all documents ("one size fits all") does not help customers in establishing display-oriented consistency comes when we analyze different approaches in designing a display-oriented XML schema. The design of the Word ML schema enables users to store in a single XML file the entire document, including content, images, table of contents, etc. This enables any person or program, for example, to send this single XML file to the wide variety of XML-enabled backend systems, such as content management tools and business workflow engines.

Sun Microsystem's StarOffice software product and Word ML use different approaches to define an application-specific, display-oriented document schema. The format in StarOffice uses many different XML files (content, styles, settings, etc.) to collectively define the document. It may prove difficult for individuals to manage a collection of multiple files such as this, as files can easily get lost or changed to render the initial document unrecoverable. It is also difficult for XML developers to understand the rationale for why certain document characteristics are placed in one XML file and other characteristics elsewhere. There is no clearly accepted segmentation of functional document features versus document formatting features.

All the files in a StarOffice document are traditionally bundled together in a ZIP file. This makes it difficult for the document information to be reused in the way that XML envisions since initially a program needs to understand how to unpack the files before it can access the XML. For example-, it should be possible to transform an XML document to HTML to enable users that do not have the XML authoring software to seamlessly view the document. However, standard XML browsers cannot use simple XSLT to transform such a zipped document to HTML because they must unzip the files first (which is not a standard XML technique). Also standard Internet-based tools like scripting engines often cannot run the executable code needed to unzip files (e.g. on a secure server or locked desktop), but they can safely parse a text file. XSLT transforms can easily be done with Word ML, however, because Microsoft's approach toward XML (and the Office file formats) is to maximize the flexibility and capability that organizations have for accessing, reusing and sharing XML document content.

Despite these apparent drawbacks, there are undoubtedly reasons that Sun's development team pursued this design path and believes it offers differentiation its customers are interested in. Perhaps the very ability to separate elements of the document can speed transmission by allowing only a necessary portion of a file to be sent across a network. Or perhaps Sun's customers have prized the ability to store files in a way that uses less memory. Regardless of the reasons, the fact that Microsoft and Sun have pursued different paths has increased customer choice without sacrificing data interoperability between Sun and Microsoft products. While each approach involves pluses and minuses, the ability to pursue different designs spurs innovation and allows customers to

choose products that best meet their needs. It is this fact that best captures why a single mandated standard for XML file formats would likely prove detrimental in the long run because if such a standard was in place, customers would not have the benefit of this innovation and choice.

8. Conclusion

As organizations around the world begin to embrace the promise of XML, there will be a significant need to engage in dialogue between the technology industry, governments, parliaments, and the many organizations that hope to deploy this technology.' While different entities will ultimately choose different paths, the collective interest in interoperability and innovation will require significant collaboration. For its part, governments and parliaments have an opportunity to create custom schema as one means to advance data exchange interoperability. In addition, popular XML techniques based on transformations (e.g. the W3C XSLT standard) enable richer document display and data exchange interoperability, where necessary, between public sector documents, authoring tools, back-ends and databases.