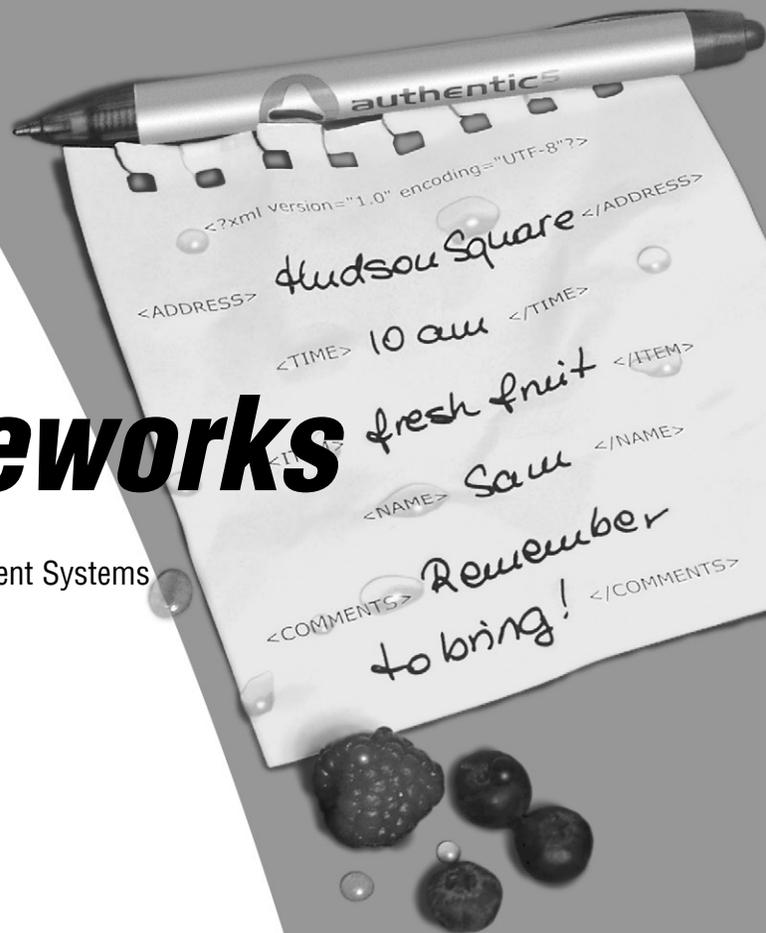


# Document Frameworks

Building Web-Enabled XML Content and Data Management Systems  
with **authentic 5™** and **xmlspy 5™**

By: Larry Kim, Technical Director, Altova, Inc.  
Member of the W3C Working Group on XQuery  
Author, the Official xmlspy Handbook (John Wiley & Sons, 2003)  
© 2003, Altova, Inc., all rights reserved.



**TABLE OF CONTENTS**

Executive Summary	3
Document Frameworks	4
Standards Based	4
Web Enabled	4
Platform and Database Neutral	4
Unifying Documents and Data	5
Cost Effective	5
The XML Content Lifecycle	5
Modeling XML	6
Editing XML	7
Using XML	8
XML Transformation	8
XML Storage	10
XML Syndication	11
Implementing Document Frameworks	12
<b>authentic 5</b>	13
Customized XML Data and Content Editing with <b>authentic 5</b>	15
Support for DTD and XML Schema	15
<b>authentic</b> Forms (Custom Template Design)	16
Browser Plug-in	16
Server and Database Integration	16
Scripting API	17
<b>authentic 5</b> Template Library for Industry Standards	17
Free Software License Model	17
Online Demonstration of <b>authentic 5</b>	17
Implementing Document Frameworks with <b>authentic 5</b> and <b>xmlspy 5</b>	20
Modeling XML with <b>xmlspy 5</b>	21
Editing XML with <b>authentic 5</b>	21
Using XML with <b>xmlspy 5</b>	21
XML Transformation	21
XML Storage	22
XML Syndication	22
Implementing Document Frameworks	22
Altova Professional Services	22
About Altova	23

## ***Executive Summary***

The ability to capture, manage, publish and exchange information assets with customers and partners in a cost effective manner is an important competitive advantage for companies in today's networked, knowledge-based economy. Development of such comprehensive enterprise information management systems represents an active technology trend in industry, and XML technologies are widely accepted as playing a vital role in enabling such systems.

This whitepaper discusses the XML Content Lifecycle and the implementation of Document Frameworks, a concept referring to XML based information management systems, which effectively streamlines the data and content gathering processes, allowing business information to be re-used and better acted upon. Altova's **authentic 5**, an XML document editor offered under free software license, is presented in a case study which investigates the design and development of loosely coupled, distributed XML content management systems. Furthermore, **xmlspy 5**, the industry standard XML Development Environment, facilitates additional customization of Document Framework applications, by providing a comprehensive feature set of utilities for working with XML.

Document Frameworks are cost-effective solutions for realizing effective information management systems which leverage open standards, existing server and database infrastructure, available know-how, and are a compelling alternative over older, high-priced, proprietary content management systems.

## ***Document Frameworks***

A Document Framework is a custom software application which captures business content and data through the use of loosely-coupled software components, servers, and repositories; all integrated through standard interfaces, enabling companies to better reuse information. Document Frameworks offer a refreshing new approach to solving the problem of business content fragmentation, and have numerous advantages over classical content management systems in the following areas:

### ***Standards Based***

Document Frameworks are built entirely on current XML technologies and Internet standards, specifically: XML Schema, DTD, XSL/XSLT, SOAP, WSDL, WebDAV, and HTTP. The use of these standards-based technologies in implementing information management systems is explained in greater detail in the following section entitled "The XML Content Lifecycle". In contrast, classical content management systems are often based on proprietary protocols and data storage formats.

### ***Web Enabled***

Document Frameworks can allow business users to create and edit information through either a desktop software application or from anywhere on the Internet, through a Browser Plug-in for Microsoft Internet Explorer.

### ***Platform and Database Neutral***

Support for Internet protocols such as HTTP and WebDAV enables Document Frameworks to be implemented in any programming language, and deployed to any server platform. Furthermore, the information captured by a Document Framework can be stored in any native XML database, XML-enabled Relational Database System, WebDAV enabled server, content management system or file system. Thus, Document Frameworks allow companies to leverage existing

servers and database infrastructure and do not require the additional purchase of expensive proprietary content management servers.

### ***Unifying Documents and Data***

Information management systems tend to be either data gathering applications, such as purchase orders and expense reports; or content editing applications, such as technical documentation and research report applications. Because XML is an ideal storing format for both data and content, Document Frameworks unify relational data with document management systems, allowing for hybrid data and document applications. Throughout this whitepaper, the term "Information Management System" is used to refer to a system which is capable of capturing and editing both business content and data.

### ***Cost Effective***

The natural outcome of building a flexible information management system is a powerful cost-effective solution, which leverages standards based technologies and existing investments in server platforms and databases. Document Frameworks allow companies to avoid the common vendor lock-in traps and pitfalls associated with conventional content management systems.

### ***The XML Content Lifecycle***

The process of creating data and content - information required for subsequent consumption or to aid in business decision making - is typically effort-intensive. The process requires organizing the information prior to creation, creating and editing the information, and formatting the information so that it may be easily used. With so much time, cost, and effort invested in information gathering, it makes sense to reduce costs by reusing information as much as possible through the use of XML technologies. XML technologies can be applied to any information gathering process through a simple process known as the XML Content Lifecycle, which involves four stages: Modeling XML,

Editing XML, Using XML, and Implementing Document Frameworks. The above steps require different technologies and resources, and are relevant to different categories of workers within an organization.

The methodology outlined in the XML Content Lifecycle applies similarly to either data or document oriented Document Framework applications, or any hybrid of the two. In the XML Content Lifecycle, organizations can move from ad-hoc, fragmented content creation systems to more structured information gathering systems, and thus truly realize the benefits of reusable content.

### ***Modeling XML***

The first step in building an XML content editing application is to model the information that is being captured and edited. The modeling process can be applied to data-oriented information, such as purchase orders or stock quotations; document-oriented information such as technical manuals or research information; or any hybrid model of both data and document oriented information.

Modeling of XML information (or documents) is done using a Meta-language (i.e. a language for describing information), which specifies a document's structure, and the rules or constraints on the value space for which each data element must adhere to in order to be considered a valid member of a particular family of documents. The description of the structure of an XML document and the rules which its contents must adhere to are collectively referred to as an XML content model. The XML content model, in turn, defines a family or class of a particular kind of document.

A content model is defined using either a Document Type Definition (DTD) or, more commonly, an XML Schema. XML Schema is the more powerful of the two Meta-languages. From a technical perspective, XML Schema is highly flexible, with support for data types and object oriented programming concepts (such as encapsulation, type derivation and polymorphism). From an industry standard perspective, XML Schema

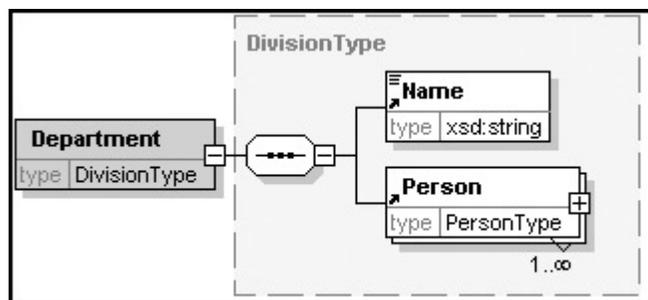


Figure 1: Developing Information Content Models using XML Schema

is recommended by the World Wide Web Consortium (W3C) and is widely supported by all major enterprise software platform and data repository vendors. A graphical illustration of an XML Schema component is shown in the following figure:

Modeling of XML content is possibly the most important first step in developing any XML content editing application, and is similar to how it is necessary to create a database schema prior to developing a database application, or how it is necessary to model classes and their relationships prior to developing any sophisticated software application using Object Oriented Programming methodologies. Dedicating attention to modeling the structure of the business information being captured and edited provides a solid foundation for subsequent steps in the XML content lifecycle.

### ***Editing XML***

Once an XML content model has been determined, the developers of the Document Framework must provide the facilities for business users to easily create and edit information conforming to the content model. The business users must not be exposed to the underlying technical implementation of the system. This is done through a simple two-step process:

1. Document framework designers must first create visual, user-interface templates or forms to customize the environment through which a business user creates and edits underlying XML content.
2. The completed user-interface templates or forms are then registered with an editing component (either a desktop software application or a browser plug-in), which enables business users to easily create and edit content conforming to the content model.

### ***Using XML***

Up until this point we have primarily been engaged in setting up a framework for capturing and editing content into XML. We have not yet realized the benefits of having chosen to adopt XML and having an XML information gathering system in place. XML has numerous benefits over other proprietary data formats in that it can be easily transformed, efficiently indexed and stored, and transmitted or syndicated via Web services technologies. While there are certainly other usages of information stored in XML, the following section focuses on these primary benefits.

### ***XML Transformation***

The eXtensible Stylesheet Language (XSL) consists of the eXtensible Stylesheet Language Transformation (XSLT), a standardized language for transforming XML documents to simple output forms such as HTML or WML, and the eXtensible Stylesheet Language Formatting Objects (XSL:FO), an XML-based language for expressing advanced document layouts, employed by many popular formats including PDF and PostScript files.

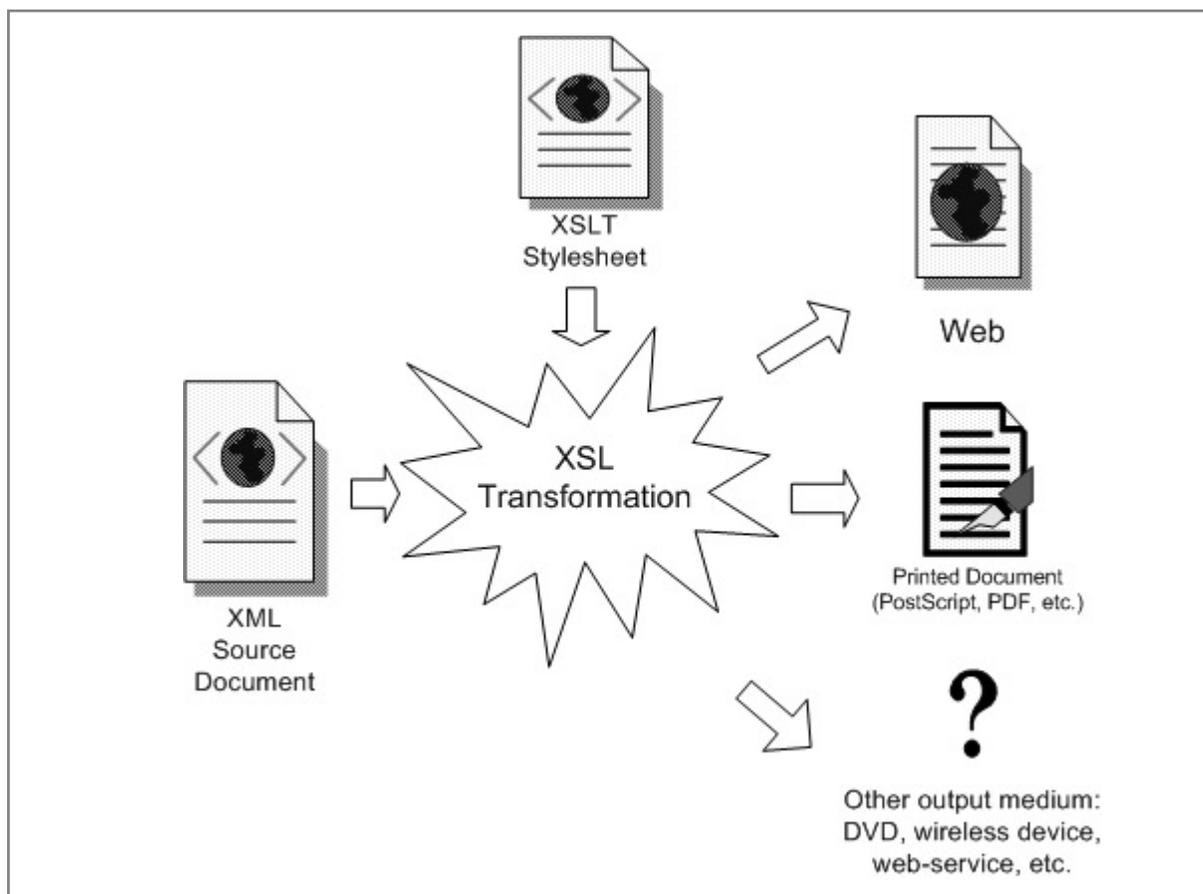


Figure 2: The XSL Transformation Process.

XSL de-couples the contents of a document from its style (i.e. the document's layout and formatting). This allows you to either change the document's style without affecting the content, or to change the content while preserving the style. The transformation process from one data format to another involves processing an XML document and an XSL stylesheet in an XSL processor, which results in the generation of a new output document as illustrated in the following figure:

An example of altering a document's style while preserving the content is multi-channel publishing. Using XSL, a single source of XML content can be published into a variety of customized output media, such as HTML, WML, PostScript, PDF, or any other information format, through the application of a stylesheet.

An example of changing a document's content while preserving the style is internationalization and localization of resource files. A corporate website could internationalize its content in different languages such as German and Japanese, simply by translating the XML content and leaving the stylesheets untouched.

### ***XML Storage***

XML content must be persisted (saved) to some database, content management system, or native XML database server. The vast majority (estimated 80%) of corporate information today is stored in relational databases, owing to the fact that the relational data storage model is very efficient at storing of tabular data. Relational databases are less efficient, however, at storing structured documents, such as technical documentation, press releases, etc. Native XML database servers excel at storing structured content because they preserve the structure and semantics of a document allowing for more efficient searches and retrievals of XML content.

Document Frameworks benefit from the best of both relational and native XML storage models, thanks to the Web-based Distributed Authoring and Versioning protocol (WebDAV), a standards-based extension to the HTTP protocol which enables collaborative editing of any information (data or content) over the Internet in a distributed fashion. As far as WebDAV is concerned, there are WebDAV clients and WebDAV Servers. Examples of a WebDAV client could include Windows Explorer, **xmlespy 5**, or **authentic 5**. Examples of WebDAV Servers include Oracle9i, Microsoft SQL Server 2000, Software AG's Tamino Server, and many others. Provided that both client and server speak the WebDAV protocol, information is easily stored and retrieved, allowing an author to lock a file, make edits, then commit or roll-back the changes. The WebDAV protocol renders older, proprietary document management protocols, obsolete, as illustrated in the following figure:

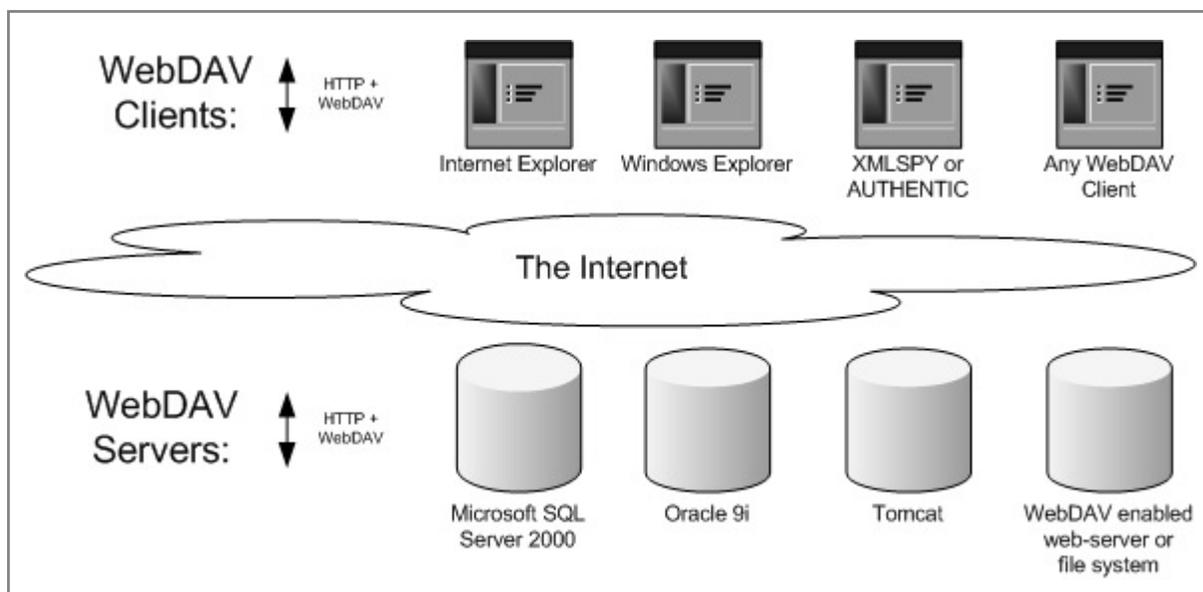


Figure 3: WebDAV makes any database or repository available for information storage in a distributed environment.

### ***XML Syndication***

A key benefit of having information (both data and content) expressed in XML is that it can be shared or consumed with the information systems of business partners and customers, on a programmatic level; herein is the basic premise behind all Web Services technologies.

Examples of syndication (or transmission) of XML data include B2B applications, financial applications, supply chain management, etc. Examples of syndication of XML content include Rich Site Summary (RSS) - a lightweight XML format designed for aggregating and syndicating headline news and other Web content. While there are fewer examples of syndication of XML content in use today, it is an area of great interest for future software applications. In summary, XML syndication requires architecting a means for a document framework to programmatically access and consume external information, as well as share (or produce) internal information with external applications.

## ***Implementing Document Frameworks***

The final stage in the XML Content Lifecycle is the implementation phase, which involves making everything work together. Up until this stage, we have been making higher level architecture decisions which affect the workflow of the Document Framework. At some point the information models, user interfaces, document editor component, XSLT engine, stylesheets, storage repository, and web services must be bound together as a unified implementation - is done in the implementation phase, and can be thought of as assembling a Lego creation since all of the constituent components adhere to standard interfaces. The glue which binds all of the various components is, of course, custom programming.

Document Frameworks can easily be implemented using any of the leading Internet application development platforms including Java 2 Enterprise Edition or Microsoft .NET, or any other platform. Free XML processing Application Programmer Interfaces (API's), and XML processing components are freely available for all programming languages, thus Document Frameworks can be implemented in Java, C#, C++, JavaScript, Perl, etc. In fact, using the Code Generator in **xmisp** 5 the majority of the infrastructure code required for such an implementation can be auto-generated directly from the XML Schema model. Furthermore, Altova even offers a free XSLT Engine (a stand-alone XSLT processor) which can be easily embedded into a server-based XML processing application. To aid with the implementation phase, Altova also offers numerous technical resources and example applications, and offers paid professional consulting services.

## authentic 5

**authentic 5** is Altova's lightweight, XML document editor application (similar to a word-processor) that lets organizations visually gather information, such as data and business content, through the use of rich, dynamic forms. It is an editing component ideally suited for the "Editing XML" stage in the XML Content Lifecycle.

**authentic 5** is designed for end-users at organizations - i.e. users who are primarily engaged in the creation of business content, or who require the ability to collect and edit important business information to perform their jobs. Typical users of **authentic 5** include reporters, analysts, lawyers, marketing professionals, order processing clerks, and a wide variety of other business users.

A key advantage of **authentic 5** over competing XML document editor products because it supports both forms-oriented data input, as well as document-oriented content editing, or any hybrid combination of the two models, while still allowing the system architect to impose constraints/structure on data entry.

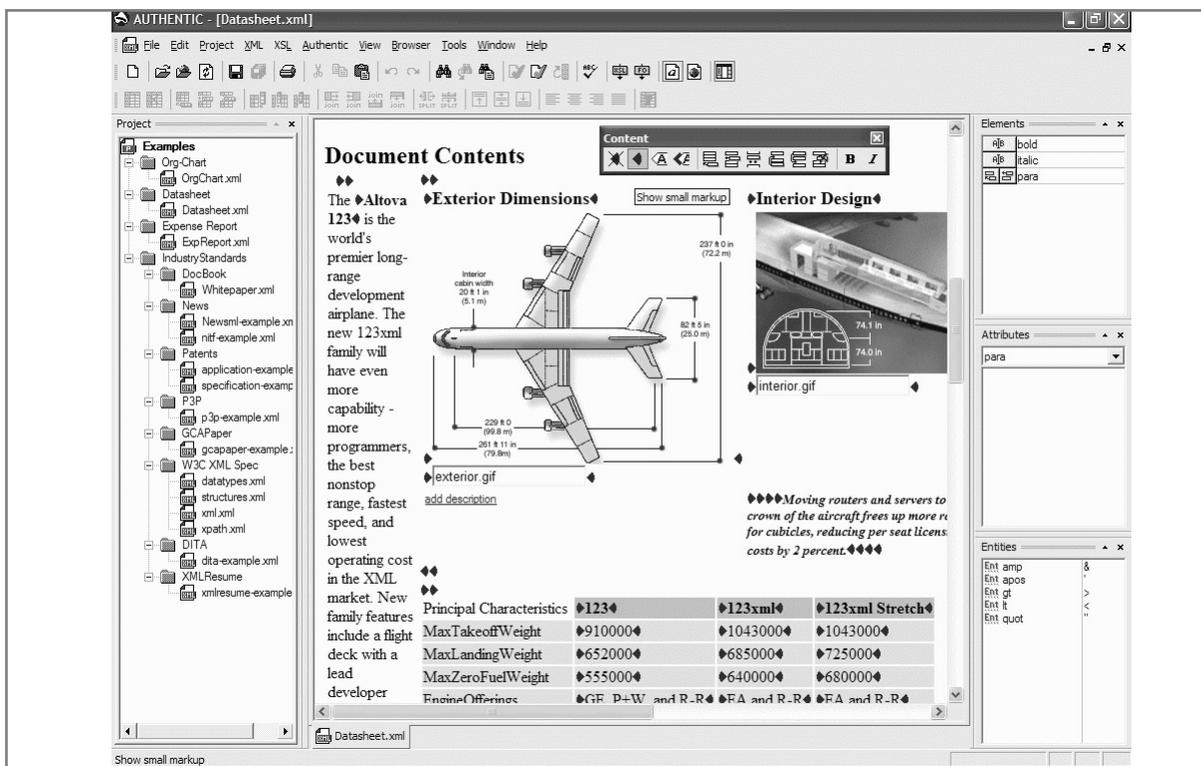


Figure4: authentic 5 Desktop Edition

Information collected by **authentic 5** is automatically stored in XML, an industry standard information format, which can then be easily validated and integrated with an organization's server systems. Storing corporate information in XML format using **authentic 5** allows companies to realize all of the benefits outlined in the "Using XML" step in the XML Content Lifecycle, including:

- >> Multi-Channel Publishing: Information gathered through **authentic 5** can be easily transformed into other formats, such as HTML, WML, PDF, PostScript, etc.
- >> Efficient Content Storage: XML content can be efficiently stored in virtually all of today's leading databases, content management servers, and XML repositories.
- >> Content Syndication: Information assets are more readily internationalized and can be transmitted to business partners or the public via Web services.

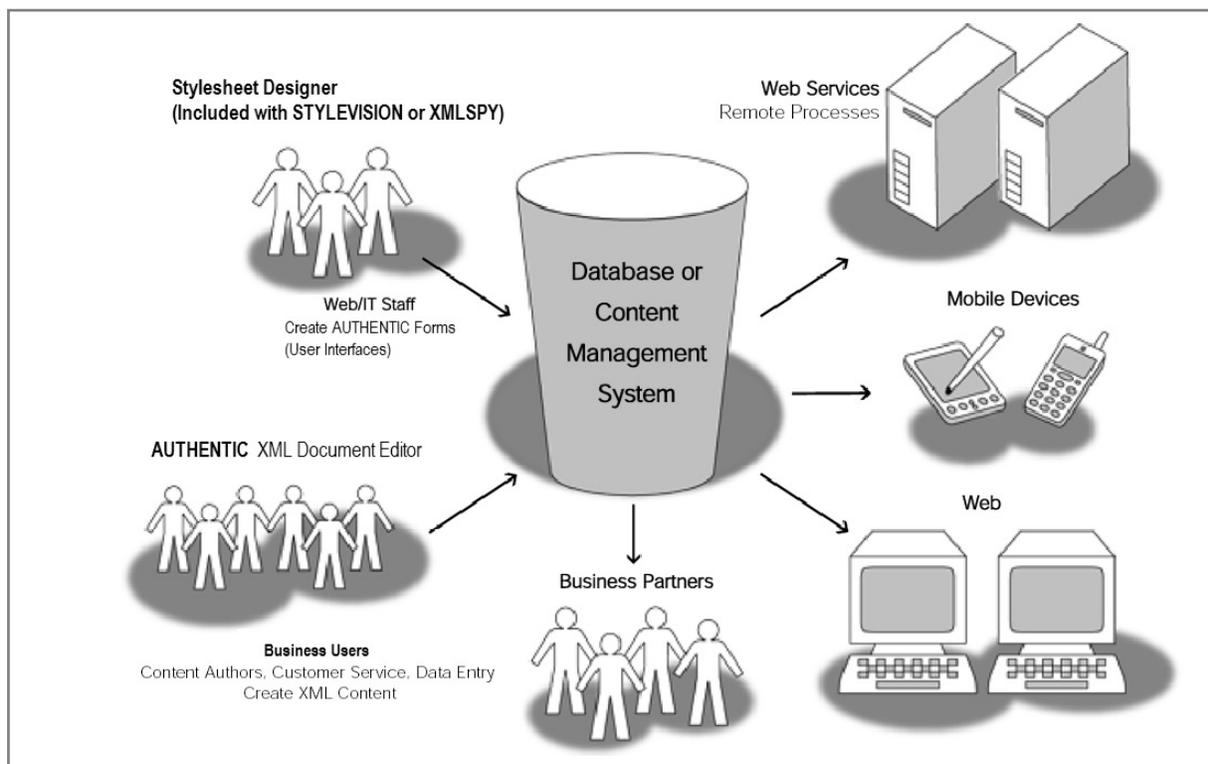


Figure 5: A Sample Document Framework Application using **authentic 5**

In summary, **authentic 5** is a key component for realizing the XML Content Lifecycle, enabling customers to efficiently aggregate information, thus allowing them to better act on that information throughout their organizations' and business processes, resulting in greater business efficiencies.

### ***Customized XML Data and Content Editing with authentic 5***

**authentic 5** is a highly customizable XML document editing component which is used today across a wide range of business applications, including financial, banking, insurance, eCommerce, publishing, defense, telecommunications, education and many other vertical industries, operating in conjunction with numerous different server and database configurations. The following sections address how **authentic 5** can be easily customized to suit the needs of any XML data or content editing application.

#### ***Support for DTD and XML Schema***

Information gathered using **authentic 5** is stored in XML, which adheres to any industry-standard content model or customer-defined data model, expressed using either Document Type Definition (DTD) or XML Schema.

### ***authentic Forms (Custom Template Design)***

**authentic** Forms are the customized templates and forms (i.e. the user-interfaces) through which a business user creates XML content in **authentic 5**. **authentic** Forms are created using the Stylesheet Designer, a visual, drag-and-drop tool which is included with both **stylevision 5** and **xmlspy 5** (Professional and Enterprise Editions).

As data and content models increase in complexity, so does the need to keep XML content editing user-interfaces and form layouts simple for your end-users. Advanced, customized visual templates for use in any XML content editing application can be made even more intuitive, through the use of global templates, and conditional statements in the Stylesheet Designer, which enables a form's layout to dynamic change the appearance of its layout and presentation, based on the data or content being keyed-in by an end user.

### ***Browser Plug-in***

**authentic 5** is available as a Desktop Application or as a Browser Plug-in for Internet Explorer. The browser plug-in is the only web-based XML content editing solution currently offered in the industry - an important feature for global companies and business users on the go. The **authentic 5** Browser Plug-in provides an intelligent Web-based data entry system which takes advantage of the rich editing possibilities of XML and XSLT, which are not possible using simple HTML forms.

### ***Server and Database Integration***

**authentic 5** features full support for standard internet protocols and file transfer interfaces, including WebDAV and HTTP. This allows **authentic 5** to be easily used in conjunction with the leading XML content repositories, including Oracle XML DB (Oracle9i), Microsoft CMS 2002 via the **authentic 5** Content Management Server Placeholder Control Edition, Microsoft SQL Server 2000, Software AG Tamino Server, IXIASOFT TextML Server, Lightspeed

---

Interactive Astoria, X-Hive/DB, iLevel Software Insite Server, NeoCore XMS, and many others - integration kits are available for free download.

### ***Scripting API***

**authentic 5** supports a rich scripting environment which gives you the ability to create and manage macros and event handlers to customize the user's editing experience. Using JavaScript or VBScript, a developer can easily create customized error messages, specify additional processing and much more.

### ***authentic 5 Template Library for Industry Standards***

**authentic 5** includes numerous built-in templates for popular XML industry vocabularies, such as DocBook, Information Text Format (NITF), News Markup Language (NewsML), Research Information Exchange Markup Language (RIXML), IBM Darwin Information Technical Architecture (DITA), Human Resources Markup Language (HR-XML), Text Encoding Initiative (TEI), P3P, USPTO, and many others.

Alternatively, any custom **authentic 5** Form created using the Stylesheet Designer, included with **stylevision 5** or **xmlspy 5** (Enterprise or Professional Edition) can be used.

### ***Free Software License Model***

It has been mentioned previously that Altova's **authentic 5** is available for use under a free software license. This is considerably less than competing XML document editors on the market, which cost between \$500-1000 (USD) per installed user; this represents a significant cost, because XML content editing applications are meant to be deployed across an entire organization. The high prices associated with all other current XML content editor product offerings becomes a significant barrier for companies wanting to develop XML content editing solutions, given the current restraints on corporate IT spending.

Altova believes that the high costs of XML content editors should not deter companies from adopting and ultimately benefiting from corporate-wide XML content applications. **authentic 5**'s free software license model affords corporate IT departments, even those under tight spending constraints, the ability to immediately start implementing custom Document Framework applications and start realizing the benefits and cost-savings associated with the reuse of information assets.

### ***Online Demonstration of authentic 5***

Altova has created several web-based samples of Document Framework applications which can be accessed from the developer section of [www.altova.com](http://www.altova.com). Current sample applications include a Product Catalog, Organization Chart, Expense Report, and News Article applications.

The Document Framework example applications consist of content-rich **authentic** Forms that were generated using **stylevision 5**, configured for use with the **authentic 5** Browser Plug-in, which will automatically install itself when you visit the website. Notice that when editing a form, repeating elements will dynamically expand themselves, a real-time XML Schema validator will highlight any invalid data entries, conditional formatting will appear based on the data that is being keyed in, and styles (such as italics or bold) will be preserved. Upon completion, click on the save button, and the generated XML file will be e-mailed to you, or saved to an Oracle9i or SQL Server 2000 database.

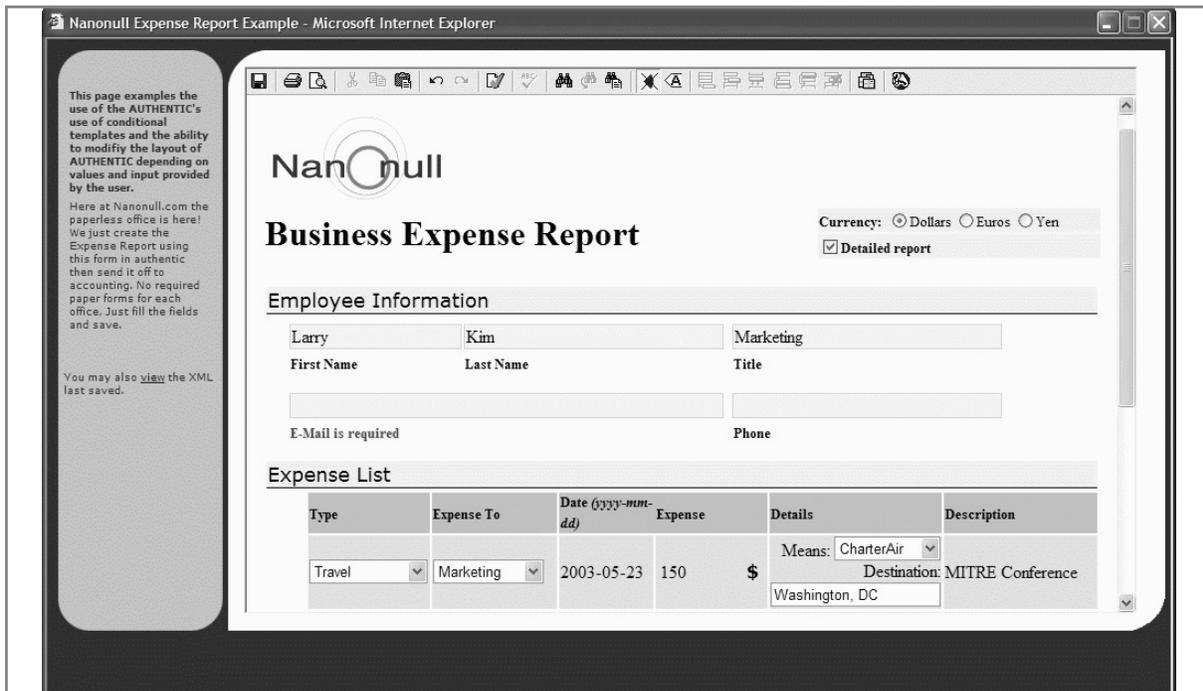


Figure 6: A sample web-based example application

## Implementing Document Frameworks with authentic 5 and xmlspy 5.

To accelerate the development of custom Document Framework examples, Altova recommends XMLSPY 5, the world's leading XML Development Environment. **xmlspy 5** can be viewed as the ultimate tool to implement the steps outlined in the XML Content Lifecycle. Even if an organization was to implement a Document Framework application using some other XML document editing component (i.e. If they elected not to use the freely available **authentic 5** XML document editor), the steps outlined in the XML Content Lifecycle would still need to be implemented, and **xmlspy 5** would still be the best tool for the job.

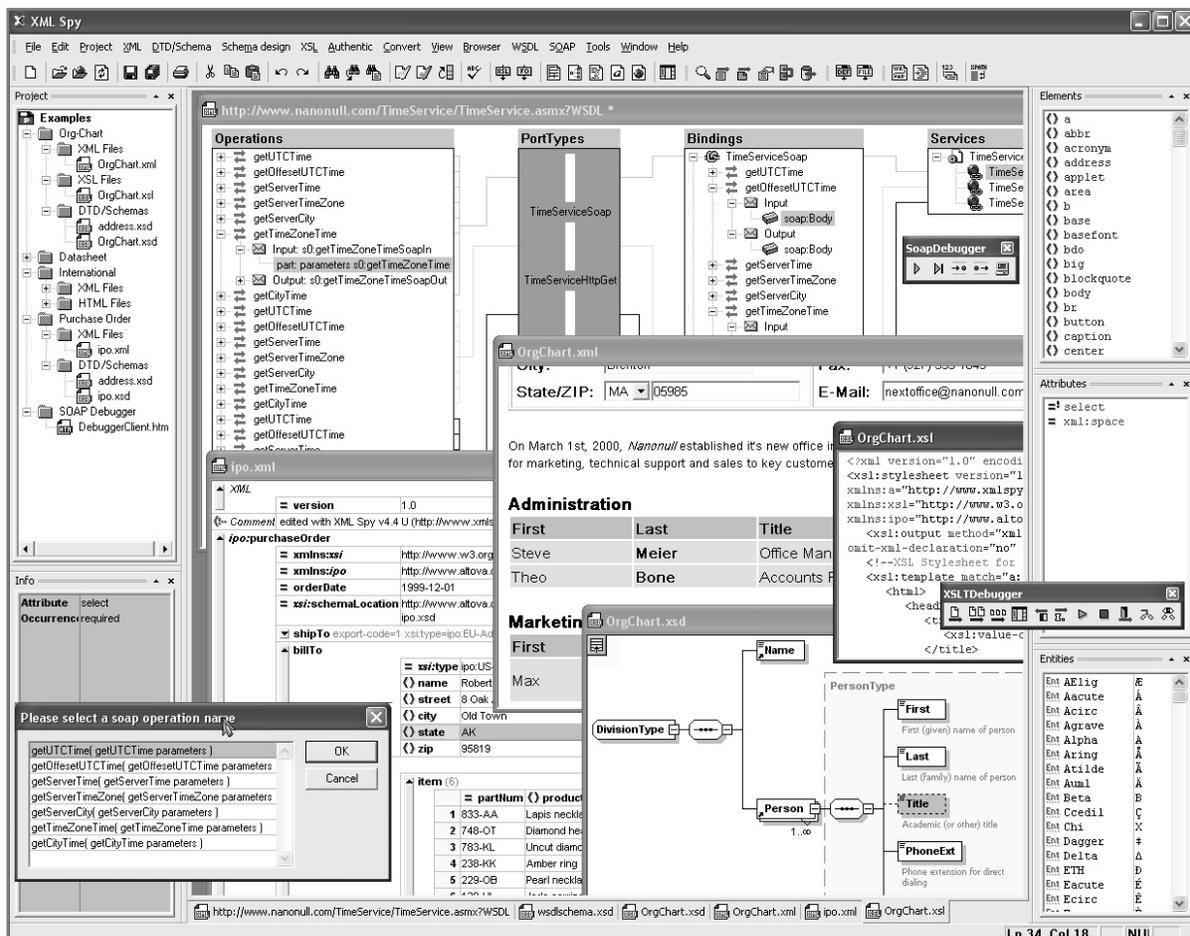


Figure 7: XMLSPY 5, the Industry Standard XML Development Environment

The easiest way to learn about **xmlspy 5** is to download a free 30-day trial of **xmlspy 5** Enterprise Edition from [www.altova.com/download](http://www.altova.com/download); the following section highlights some of the features and where they would be used throughout the XML Content Lifecycle. The complete **xmlspy** feature list can be viewed at <http://www.altova.com/matrix.html>.

### ***Modeling XML with xmlspy 5***

- Graphical XML Schema Editor
- DTD Editor
- Content Model Conversion Utilities
- XML Schema Generation Tools (From Databases, Instance Documents)
- XML Schema Documentation Generator

### ***Editing XML with authentic 5***

- XSLT Stylesheet Designer
- XSL:FO Stylesheet Designer
- Well-formedness checking
- Validation (DTD & XML Schema-based)
- Dynamic Forms for context-sensitive document editing
- Spell-Checking
- CALS/HTML Table Support

### ***Using XML with xmlspy 5***

Too often, not enough emphasis is placed on actually realizing the benefits of having adopted XML as an information storage format. **xmlspy** enables developers realize the benefits of XML by providing powerful utilities to transform, save, and transmitted XML documents, through the following utilities:

### ***XML Transformation***

- XSLT Debugger
- XPath Analyzer
- XSL:FO Editing
- Altova XSLT Engine

### ***XML Storage***

- Database Import / Export (ODBC / ADO)
- Software AG Tamino Server Administration
- Oracle XML DB Server Administration
- Microsoft SQL Server 2000 SQLXML editor

### ***XML Syndication***

- WSDL Editor
- SOAP Tester / Debugger

### ***Implementing Document Frameworks***

- Java/C++/C# Code Generation
- Technology partnerships and integration with the leading database, server and software development tools providers.

### ***Altova Professional Services***

Altova Professional Services provides world class consulting on XML technologies, systems integration best practices, and the development of custom Document Framework applications. Altova Professional Services work in conjunction with Altova's technology partners and customers to bring their complex projects to successful conclusions.

Altova's expertise of XML technologies is the foundation of the professional services group. The same company that created **xmisp**, the industry standard XML Development Environment is the perfect place to turn to when the question is how to develop XML based applications. Each of our experts is prepared to listen to your needs, clearly communicate the possibilities, and roll up their sleeves to work with your team on a high quality solution.

Our consultants' focus is at all times on knowledge transfer. For larger groups, Altova professional services group provides on-site XML centered training. For more information contact:  
professional-services@altova.com.

### ***About Altova***

Altova, Inc. is a leading provider of XML software solutions with offices in Beverly, Massachusetts and Vienna, Austria. The privately held company was founded in 1992 and has been actively involved in the XML market from the early conception of Extensible Markup Language. Altova's **xmlspy 5** is the leading choice of Fortune 500 and Global 1000 companies in use by over a Million registered users today. Altova's **xmlspy 5** product line is the world's best-selling XML tool and has won the leading industry awards including PC Magazine's Editor's Choice Award and Best Product of 2002. Altova is a technology partner with leading software companies including Microsoft, Oracle, BEA, IONA, and Software AG. Altova is a member of the W3C and WS-I. Visit Altova on the web at <http://www.altova.com>.