

Altova, Inc., The XML Spy Company  
900 Cummings Center, Suite 314-T  
Beverly, MA, 01915-6181, USA  
Tel: 978-927-9400  
Fax: 978-927-7447  
Web: [www.xmlspy.com](http://www.xmlspy.com)

ALTOVA

[www.altova.com](http://www.altova.com)

xml spy

# Document Frameworks

*Unifying XML Content Management and Database Systems for the Internet*

**Table of Contents**

<b>Executive Summary</b>	<b>3</b>
<b>History of Content Management</b>	<b>4</b>
<b>Requirements for Effectively Managing Content</b>	<b>5</b>
<b>Document Framework</b>	<b>6</b>
XML Schema	6
WebDAV	6
XSL/XSLT	8
Unicode	8
<b>Document Framework Design – Advanced XML Application Development</b>	<b>9</b>
Schema Modeling	9
Data Flow & Process Modeling	9
Transformation Modeling	9
Implementation	9
<b>XML Spy 4 Suite</b>	<b>10</b>
XML Spy 4 IDE	10
XML Spy 4 Document Editor & Browser Plug-In	12
XML Spy 4 XSLT Designer	14
Document Framework Deployments	15
<b>Online Demonstration of Document Framework</b>	<b>16</b>
<b>About Altova</b>	<b>17</b>

## ***Executive Summary***

**Companies** are spending millions of dollars each year to improve corporate IT software infrastructure with the hopes of gaining competitive advantages through enhanced customer service, reduced costs, and integrated business systems. A recent focus of corporate IT spending has been in automating transaction processing capabilities and providing enhanced web-based user interfaces - regrettably, less focus has been placed on the more informal, but equally critical task of streamlining knowledge gathering & content management processes.

**Industry movement** towards developing comprehensive enterprise information management systems is currently one of the most active technology trends in today's economy, and XML has been clearly identified as the key technology to enable such systems. This document discusses the challenges, benefits, and other important considerations to address when developing document frameworks, a concept referring to providing universal access to information stored in any underlying database or proprietary content management system throughout a corporation. The XML Spy 4 Suite is presented as a case-study on developing loosely-coupled, distributed XML content management systems based on open industry standards: XML Schema, XSLT, WebDAV and Unicode.

**The benefits** of implementing corporate-wide document frameworks include reduced costs in creating, maintaining, and distributing information to clients across a variety of different media, and superior retrieval and sharing of business-critical information.

Especially in today's difficult economic climate, the success of corporations will be largely a function of their abilities to cost-effectively capture, manage, publish and exchange its information assets, internally as well as externally, with customers and partners. Document frameworks are ideally suited for this task, because they provide a unique combination of leveraging existing database systems, present developer know-how, and open standards to produce a content and knowledge management solution tailored to the very needs of the enterprise, which results in a very low TCO (total cost of ownership).

## ***History of Content Management***

**The past decade** has introduced a multitude of new media for delivering content to customers and partners. The Web, wireless devices, web-services, e-books, DVD's, print - and still new media being developed every year - they each require a process for creating, updating and delivering information to the customer.

In most organizations, corporate knowledge is scattered across personal computers and a variety of server products including relational databases, XML repositories, and proprietary content management systems, all in their respective underlying file formats. Corporations have a great deal of resources invested in existing database systems, and are only willing to adopt technologies which would be compatible with existing investments.

The above mentioned challenges can have potentially devastating effects on corporate productivity. There is often no meaningful way of searching and retrieving information, resulting in loss of information, as well as duplicated efforts when creating new content. Because multiple copies of similar data are located across the network, it is difficult to synchronize the maintenance of content, resulting in outdated or erroneous data.

In an attempt to develop solutions to address the problems of managing content, many companies have resorted to purchasing costly proprietary document management systems that are unable to access the vast majority of enterprise data currently stored in database records.

**New standards-based technologies** have emerged in recent years which can finally address the problems of creating and maintaining XML content while dramatically reducing costs; these technologies are collectively referred to as document frameworks.

## ***Requirements for Effectively Managing Content***

Any system seeking to address the challenges of managing information across a corporation must address four key requirements.

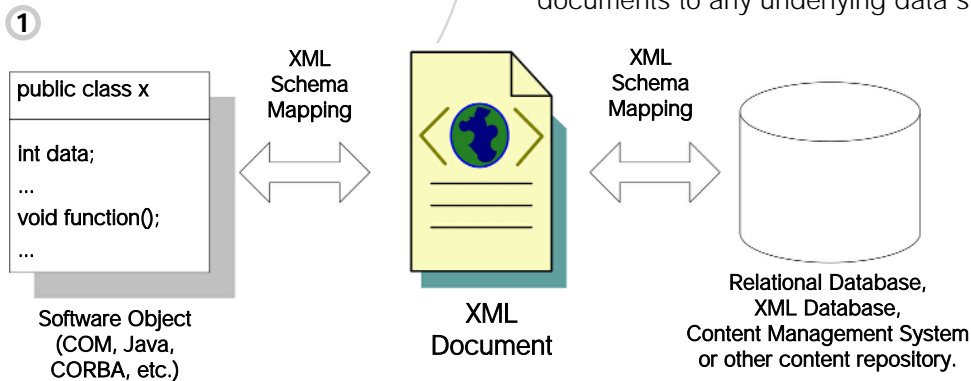
- ① **Creating new content:** The system must provide the facilities for authors to create content. The content authors must not be exposed to the underlying technical implementation of the system.
- ② **Organizing & Integrating content:** The system must provide saving capabilities to any underlying storage facility – Relational Database, XML Repository or proprietary Content Management System. The system must also support concurrent editing and updating of documents, including versioning and source control options. Finally, the system must provide content access control functionality to insure that different users have different privileges when working with content.
- ③ **Intelligently retrieving content:** Different database and content management systems are optimized for certain types of information retrieval. Consider the differences between a keyword search-engine, versus a hierarchical yellow-pages directory listing – both searching methods have relative strengths and weaknesses, thus, the system must support all enhanced content retrieval mechanisms, preserving both the context and structure of the data.
- ④ **Transforming content:** Must provide the visual tools required to easily design input-templates used to collect information, as well as the output style sheets used to transform content to any output format (print, web, wireless, DVD's, etc.)

## Document Frameworks

A document framework is a collection of software products and components used for developing loosely-coupled systems which manage corporate content for use over the Internet. XML, XSLT, WebDAV and Unicode are the technological foundation of a document framework.

**XML schema** is the World Wide Web Consortium's (W3C) official data definition language; it addresses many shortcomings associated with Document Type Definitions (DTD), and has industry support from all major software corporations (Microsoft, Oracle, Software AG, Altova, etc).

XML Schemas serve a critical role in the management of content because it can act as an intermediate translation bridge between databases and software objects. Specifically, XML Schema's object-oriented features allows for mapping between XML documents and objects written in any object-oriented programming language. XML Schema also preserves data-type information and parent/child relationships, which allows for mapping of XML documents to any underlying data store.



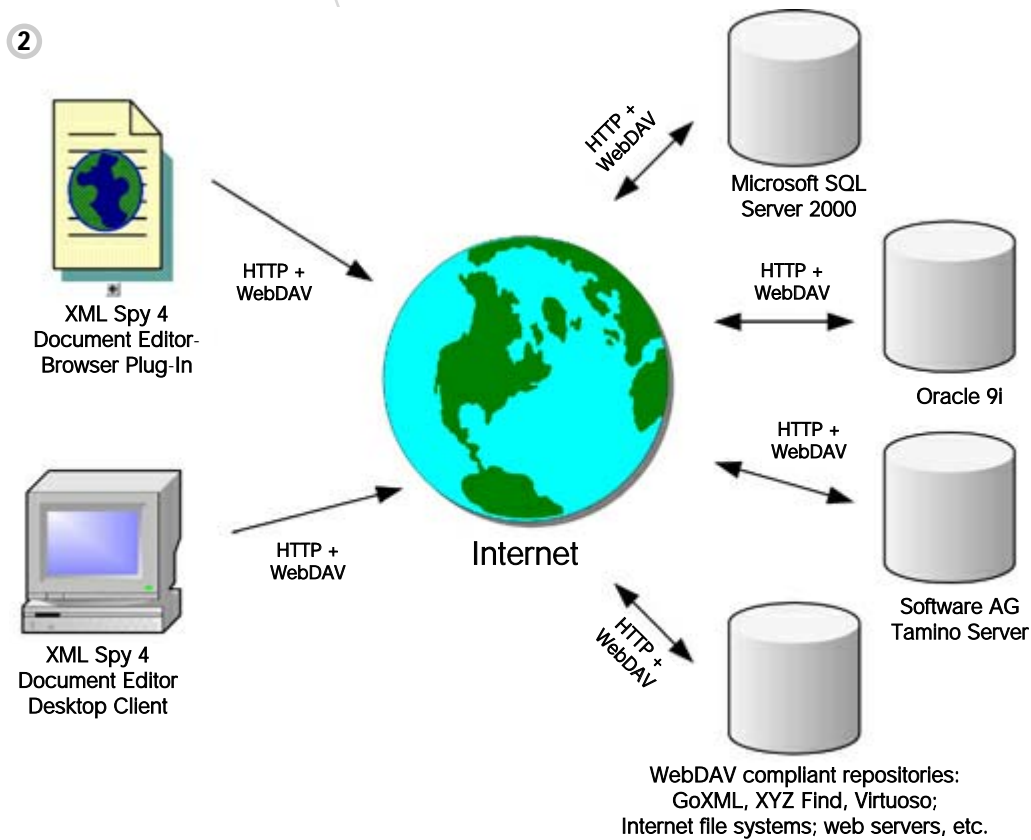
**Figure 1**

*XML Schemas enable translations from Objects to XML Documents, and from XML Documents to databases (and vice-versa).*

**WebDAV** stands for Web-based Distributed Authoring and Versioning. It is a standardized set of extensions to the HTTP protocol - the core of the World Wide Web - which allows users to collaboratively edit and manage files on remote web-servers.

WebDAV has been steadily gaining industry adoption from top software vendors (Microsoft, Oracle, Software AG, etc), with implementations making their way into popular browsers, editors and servers. WebDAV is poised to transform how users interact with what is currently for the most-part a read-only World Wide Web. Whereas the Internet historically has been limited to display and download capabilities, WebDAV embedded in software and systems promises to turn the Internet into a writeable medium capable of supporting collaboration and distributed file sharing. The protocol, which was developed by the Internet Engineering Task Force (IETF) standards process, has features that include locking and unlocking capabilities to prevent the overwriting of changes, versioning control, and Secure Sockets Layer (SSL) support to ensure proper security.

**The XML Spy 4 Suite** The XML Spy 4 Suite implements the WebDAV repository interface to ensure compatibility with a wide range of server products, including Oracle 9i, Microsoft SQL Server 2000, Software AG Tamino Server, Global XML's GoXML Server, and many others. WebDAV technologies effectively enable companies to break free of an age-old vendor-lock-in trap employed by many proprietary XML content management systems requiring that customers purchase both the content creation tools and back-end server products from the same vendor. Using the XML Spy 4 Document Editor Browser Plug-In, content authors can collaboratively create & edit information stored in any WebDAV repository on the Internet, as illustrated below:



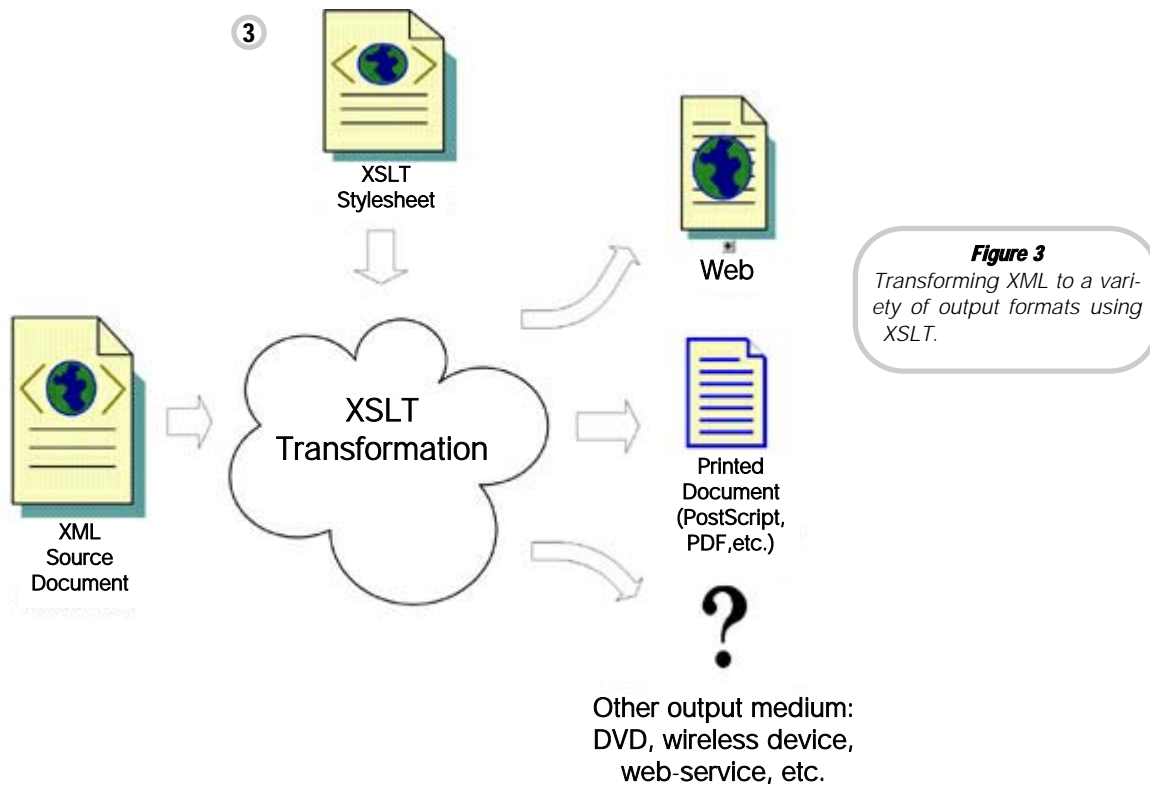
**Figure 2**

*Using WebDAV, content authors have distributed access to virtually any underlying database or content management system.*

### XSL/XSLT

The eXtensible Stylesheet Language (XSL) and the eXtensible Stylesheet Language Transformations (XSLT) are standardized languages for transforming XML documents to a different output form. Using XSL, content saved in an XML format can be transformed into any output media, by applying an XSLT stylesheet.

The eXtensible Stylesheet Language also includes a sub-section known as XSL:FO, which stands for Formatting Objects. XSL:FO is an XML-based language for expressing production-quality document layouts, employed by many popular formats including PDF and PostScript files.



**Figure 3**  
Transforming XML to a variety of output formats using XSLT.

**Unicode** is a character encoding which provides a unique number for every character, regardless of the platform, program or language you are using. This is of critical importance in promoting interoperability in today's global economy. The Unicode standard has been adopted by all major software companies, including Altova. XML Spy has fully supported Unicode from day one, which includes complete character sets for over 170 languages in 39 different scripts (or writing systems) such as Arabic, Hebrew, Chinese, Japanese, and Korean. Many content management systems or SGML-era legacy applications do not support Unicode, or claim that they will support Unicode as a "new" feature; this has resulted in failed attempts to internationalize business systems.



## ***Document Framework Design – Advanced XML Application Development***

The document framework design process is referred to as Advanced XML Application Development (AXAD) – it is an XML-centric design methodology which can be used for building document framework applications across any industry. It is a simple four-step process.

### **Schema Modeling**

XML Documents require a content model, commonly expressed in the form of the W3C's XML Schema. Developing an XML Schema is an iterative process which involves initial requirements analysis, use-cases, as well as examination of existing data schemas. Once the XML Schemas are designed, additional refinements are required to map all of the elements of your XML Schema to an underlying database (relational or XML-based) or content management system.

### **Data Flow & Process Modeling**

In every document framework deployment, there are always content authors and content consumers. Content authors are non-technical domain experts whose primary task is to create content on any subject. Content consumers are typically customers, partners or internal departments who make use of XML content. The flow of information gathered by a document framework must be modeled – who is the content author? What is the underlying data store? How is the information being transported to the database? These issues must all be addressed in this stage.

### **Transformation Modeling**

XSLT has a two-fold critical role in designing a document framework; it is required to design both the input templates that are used by content creators, and the output stylesheets that are required by the content consumers. If the desired output is a printed document (e.g. PDF or PostScript file), then XSL:FO must be employed. The developer must design the XSLT stylesheets to fit the data flow and process model determined earlier.

### **Implementation**

The business logic and user-interface of a document framework application must be custom developed. Since the components of a document framework consists of loosely coupled, lightweight components adhering to standard interfaces, a document framework can easily be implemented using any of the leading Internet application development platforms including Java 2 Enterprise Edition, Microsoft .NET, Web-Services, Oracle Application Server, or any other platform. Free XML processing Application Programmer Interfaces (API's) are available for all programming languages, thus, document frameworks can be implemented in Java, C#, C++, JavaScript, Perl, etc.

## XML Spy 4 Suite

The XML Spy Suite is the ultimate software development tools suite for Advanced XML Application Development! The XML Spy 4 Suite consists of the XML Spy 4 Integrated Development Environment (IDE), the XML Spy 4 XSLT Designer, and the XML Spy 4 Document Editor.



**Figure 4**  
The XML Spy 4 Suite product line.

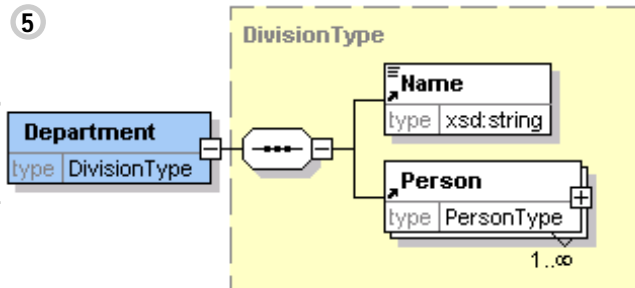
### XML Spy 4 Integrated Development Environment (IDE)

The XML Spy 4 IDE is the world's first and leading IDE for XML. In this section, we highlight several features which are used throughout the Advanced XML Application Development process to assist with implementing document frameworks: XML Schema editing, XSL/XSLT editing, and web-services testing & debugging.

The XML Spy IDE includes an XML Schema editor, which can be employed for *Schema Modeling* – the first step in designing a document framework. The XML Spy 4 IDE includes a full-featured graphical XML schema editor which supports many automatic XML Schema generation capabilities, which greatly accelerate the schema modeling process; some of the possibilities are listed below:

- 1 The XML Spy IDE can inspect one or more XML instance documents – typically these will be use-cases – and automatically infer the underlying XML Schema.
- 2 The XML Spy IDE can connect to any relational database (through ODBC or ADO) and automatically generate the corresponding XML Schema, preserving data-type information, relationships, and other data restrictions.
- 3 The XML Spy IDE can convert older content models (DTD, DCD, or BizTalk Schema) to the most current official W3C XML Schema final recommendation.

Further refinements can easily be applied to an XML Schema, using XML Spy 4 IDE's schema editor, which supports graphical editing of XML schemas. The XML Spy IDE's schema editor is the only tool to support XML schema-to-database mapping tools through the use of various 3rd party XML Schema extensions – including Oracle 9i XDB and Microsoft SQL Server 2000 XML Schema Extensions.



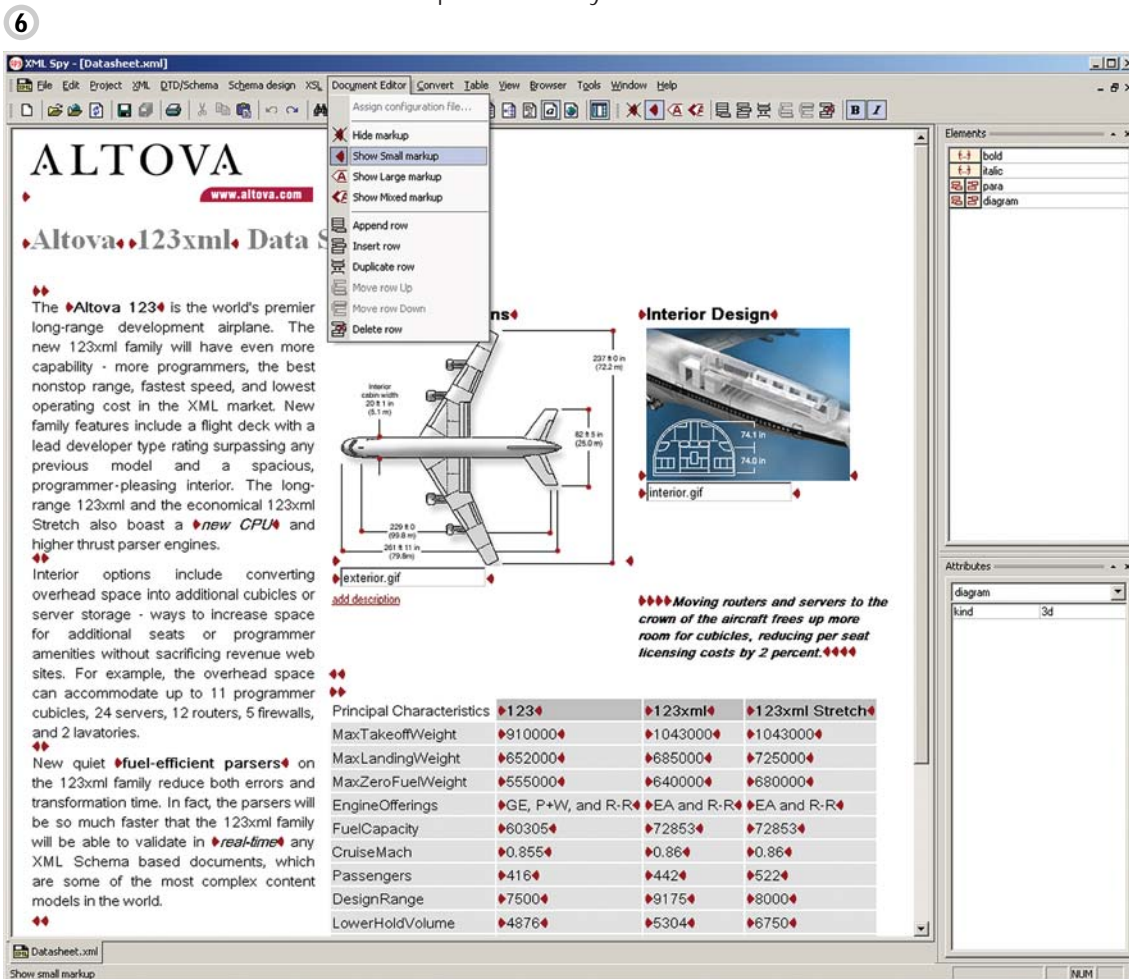
**Figure 5**  
The XML Spy IDE's schema editor allows for easy visual representation of complex elements.

The XML Spy IDE contains many features to support *Transformation Modeling* – the third step in developing document frameworks. To assist with the development of XSLT stylesheets, the XML Spy IDE supports single-click transformations, graphical XPath query generation and visualization, syntax help & code-completion for all XSL elements & attributes, support for XSL:FO to generate and preview popular output file types (PDF, PostScript, etc), and much more.

The XML Spy 4 IDE supports the development, testing & debugging of web-services. *Data Flow & Process Modeling* of sophisticated document frameworks could potentially include web-services as a key mechanism to expose document frameworks to partners and customers.


## XML Spy 4 Document Editor with Browser Plug-in

The XML Spy 4 Document Editor is a light-weight editor used to create XML content in a document framework system. It is an innovative new visual approach to writing XML documents, exposing the end user with a word-processor-like interface, and not the complicated underlying XML syntax. Using the XML Spy 4 Document Editor, information gathered by employees across the company is immediately saved to an underlying XML format, ensuring that information is valid and does not become lost or unusable. The XML content can be saved to an underlying database or content management system, to be reused and repurposed for any reason at a later time.



**Figure 6**

The XML Spy 4 Document Editor is shown above; it supports word processor like free-flow WYSIWYG text editing, form-based data input, graphical elements, presentation and editing of arbitrary repeating XML elements as tables, real-time validation, and consistency checking using XML Schema. The document template in the above screenshot was created using the XML Spy 4 XSLT Designer.



**The XML Spy 4 Document Editor** can be deployed in a document framework as a stand-alone desk-top application or browser plug-in. The decision concerning which product-variation to use is determined through *Data Flow & Process Modeling*, which is the second step outlined in the Advanced XML Application Development process. As a guideline, if content authors will always be located within the company network, then the Document Editor can be deployed as a desk-top application – however, in today's global economy, it is often the case that content authors are located anywhere across the Internet, and as such, require easy Internet access to the document framework system. In this case, The XML Spy 4 Document Editor Browser Plug-In is a unique solution that allows live XML content editing from a web-browser. The XML Spy 4 Document Editor Browser Plug-in is self-installing (similar to a Macromedia Flash, or Apple QuickTime plug-in), which dramatically eases deployment and reduces total cost of ownership; it is the only browser-based solution for rich-content editing currently offered in the industry.

The XML Spy 4 Document Editor efficiently captures information as it is being created, preserving the context in which it was produced, and the relationships between it and other existing corporate data. Through WebDAV, the XML Spy 4 Document Editor provides integration with the leading XML databases, relational databases, and content management systems.

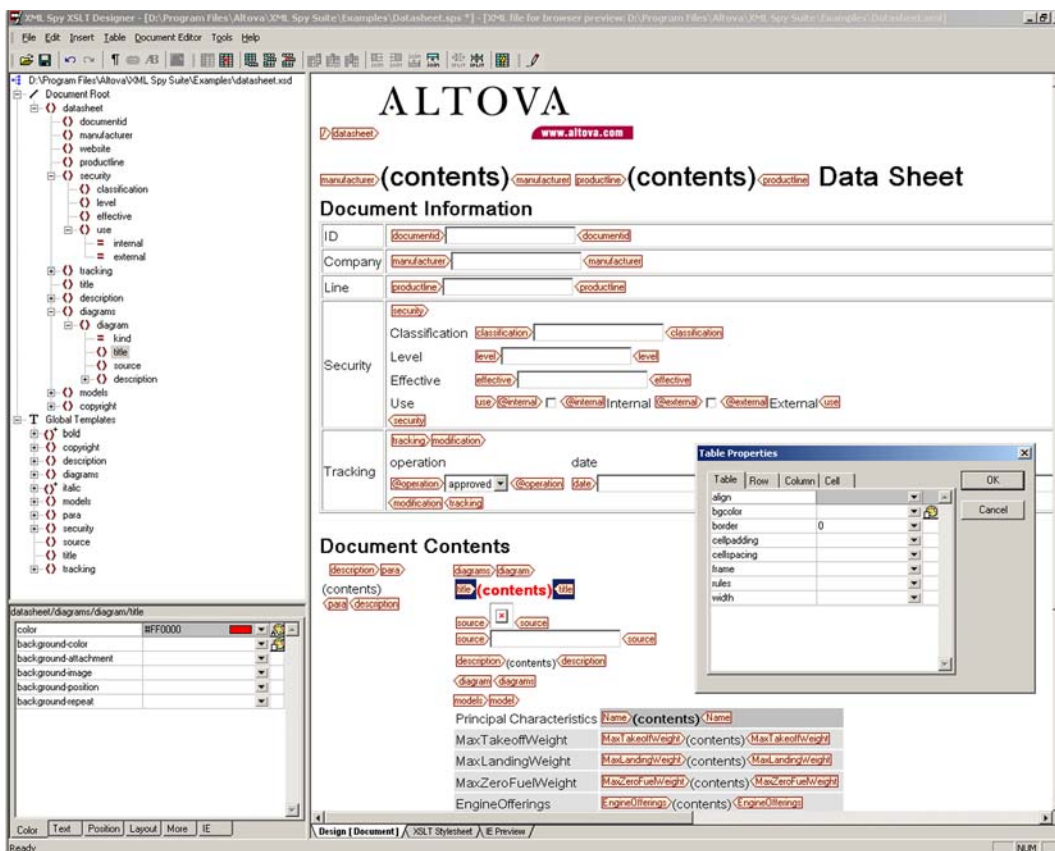
## XML Spy 4 XSLT Designer

Writing even the most basic XSLT stylesheets by hand is a truly daunting task, requiring an understanding of XSL elements, the XPath query language, and complicated rules-based document processing models. Yet, without the ability to transform and render your XML content to a desired output, the decision of adopting XML remains for the most part an academic exercise.

The XML Spy 4 XSLT Designer automates writing of complex XSLT stylesheets through an intuitive, drag-and-drop user interface; it is ideal for use throughout the *Transformation Modeling* process, the third step outlined in the Advanced XML Application Development methodology.

**Through a powerful GUI,** a user can drag and drop XML data elements corresponding to an XML Schema or DTD (in the left window) into the main design window (the right window), adding descriptive text and presentation tags such as tables, hyperlinks and graphics. The resulting XSLT stylesheet is automatically generated and can be previewed by clicking on the browser-preview tab.

7



**Figure 7**

The XML Spy 4 XSLT Designer main screen: Easy-to-use tool makes writing XSLT style sheets simple.

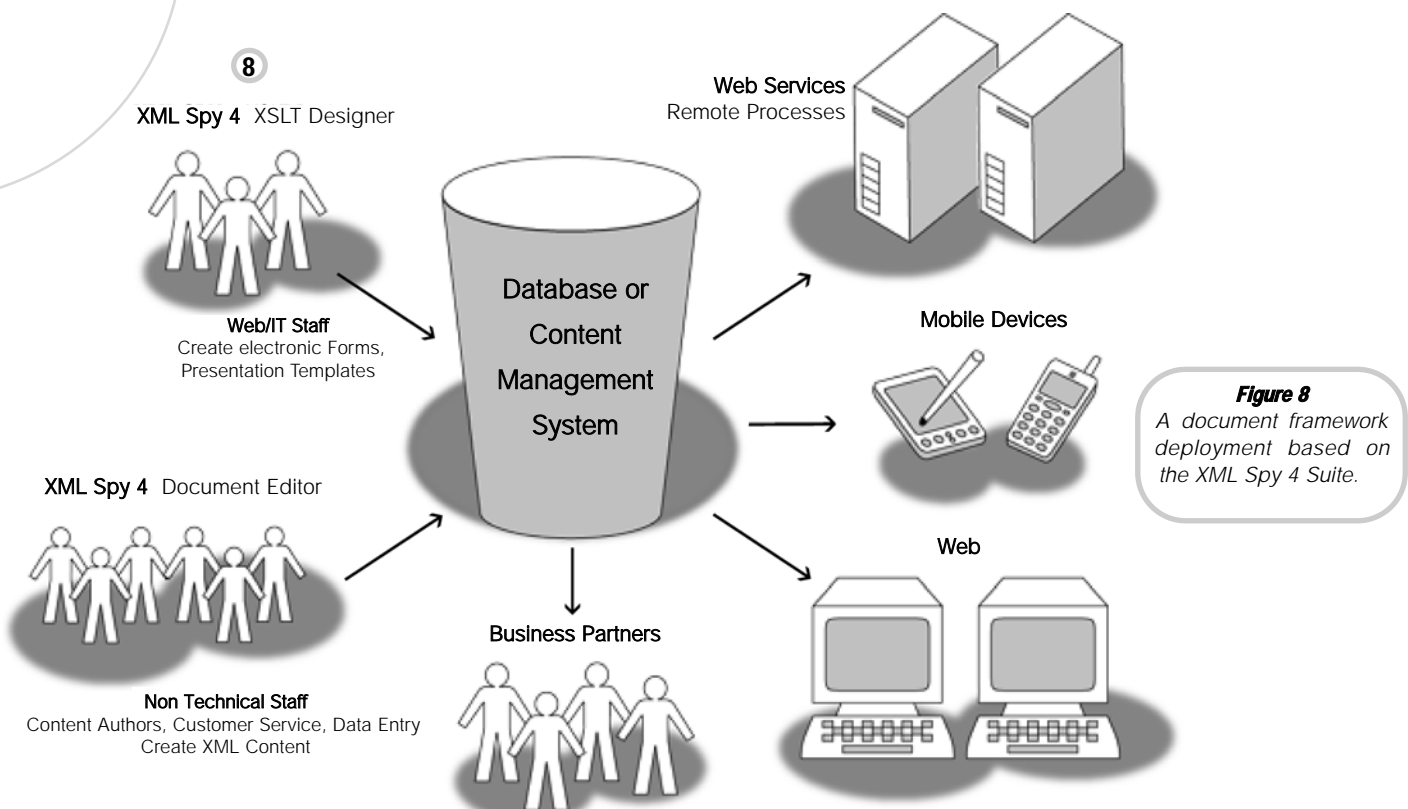
The XML Spy 4 XSLT Designer is a clear and easy to use tool. Unlike complicated, proprietary content management systems, the intuitive user-interface allows any web-designer (even with little or no understanding of XSLT) to leverage existing skills and design advanced stylesheets.

## Document Framework Deployments

The interactions between the XML Spy 4 XSLT Designer and XML Spy 4 Document Editor in a document framework deployment are summarized below:

- 1 The XML Spy 4 XSLT Designer is used to develop the rich dynamic forms and layouts which configure the XML Spy 4 Document Editor; it is also used to develop the presentation templates which transform XML into the desired output format.
- 2 The XML Spy 4 Document Editor & Browser Plug-in can be deployed to the web, enabling organizations employing teams of content authors, spanning multiple geographic locations to collaboratively create, retrieve and edit content over the Internet.
- 3 The XML data generated by the XML Spy 4 Document Editor is saved into a content management system or database such as Microsoft SQL Server 2000, Oracle 9i, Global XML's GoXML DB, or Software AG Tamino Server, where it can later be accessed by any web-browser, handheld device, web-service, or integrate directly with customers and suppliers.

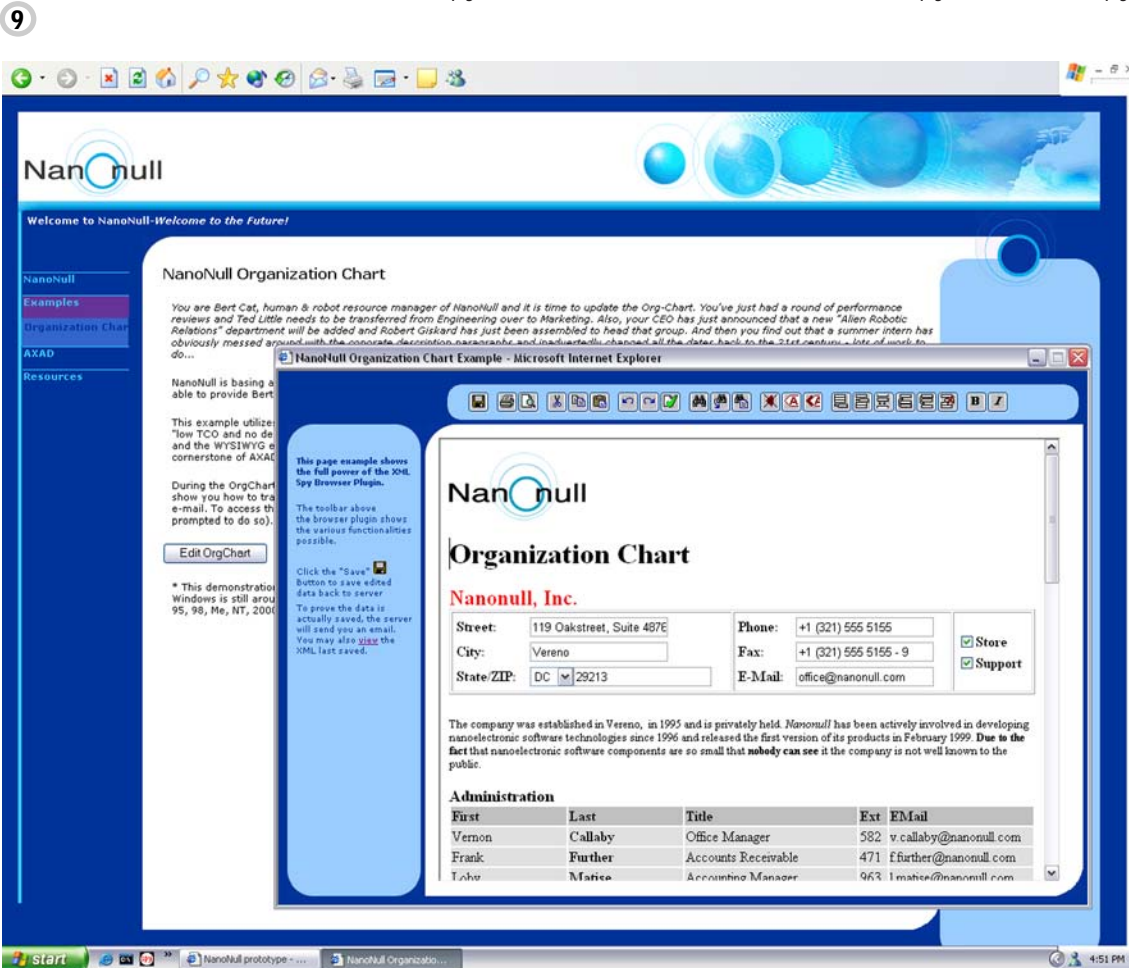
The following figure illustrates the use of XML Spy 4 Suite in implementing document frameworks throughout the enterprise.





## Online Demonstration of a Document Framework

Several sample document framework applications, complete with source code and documentation, are available online in our “NanoNull” example website at [www.nanonull.com](http://www.nanonull.com) (requires Internet Explorer 5.5 or 6.0). The demonstration consists of content-rich electronic forms that were generated by the XML Spy 4 XSLT designer. The forms are edited using the XML Spy 4 Document Editor Browser Plug-in which will automatically install itself when you visit the website. Notice that when editing a form, the tables will dynamically expand themselves, and a real-time XML Schema syntax checker will highlight any invalid data entries, 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. To start developing with XML Spy Suite, download a free evaluation copy at [www.xmlspy.com](http://www.xmlspy.com) !



**Figure 9**  
Sample Document Framework application implemented using XML Spy 4 Document Editor Browser Plug-In.



## ***About Altova***

Altova produces and markets XML Spy 4 Suite, the ultimate development tools suite for Advanced XML Application Development. Founded in 1992 (formerly Icon Information-Systems), Altova has offices in Beverly, Massachusetts and in Vienna, Austria. Altova is a major technology partner with the world's leading software companies including Microsoft, Oracle, Software AG Inc. XML Spy is used by over 500,000 registered users across 25,000 companies worldwide, and is the leading choice of Fortune 500 and NASDAQ 100 companies. Altova is a member of the W3C.



---

**ALTOVA**

[www.altova.com](http://www.altova.com)