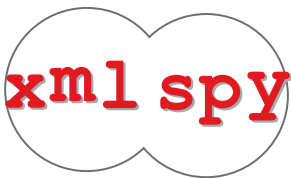


# Streamlining content creation, retrieval, and publishing on the Web

*Using TEXTML Server and XML Spy 4 Suite in an integrated,  
Web publishing environment.*

 TEXTML  
</SERVER> xml spy

## White Paper

---

July 2002

# Table of Contents

<b><u>XML</u></b>	<b><u>4</u></b>
<b><u>TEXTML SERVER</u></b>	<b><u>5</u></b>
<b><u>XML SPY 4</u></b>	<b><u>8</u></b>
<b><u>THE XML SPY INTEGRATION KIT</u></b>	<b><u>9</u></b>
<b><u>SAMPLE SYSTEM ARCHITECTURE</u></b>	<b><u>11</u></b>
<b><u>CONTENT CREATION</u></b>	<b><u>12</u></b>
<b><u>STORAGE</u></b>	<b><u>14</u></b>
<b><u>PUBLISHING</u></b>	<b><u>14</u></b>
<b><u>CONCLUSION</u></b>	<b><u>16</u></b>

## Introduction

It is estimated that 80% of a company's knowledge is locked up in individual computers. In most cases, these computers, acting as heterogeneous knowledge repositories, are linked by a common network, while the knowledge remains inaccessible to those that could benefit from it the most. Most of this knowledge is in the form of documents written by individuals such as reports, emails, presentations, which are typically unstructured or semi-structured in nature and are not easily shared. The information in these documents is rendered unusable due to this fragmentation.

According to an Intellor Group<sup>1</sup>, Inc research paper, 38% of respondents to a survey on structured vs. unstructured content say that their organization expends 90% of IT resources on structured vs. 10% on unstructured data. Of course it is natural that companies spend capital where there is a tangible ROI. Managing e-business transactions, where information comes in the form a highly structured data, gives immediate feedback. Other equally important, yet less easily measurable benefits such as efficiency gains due to improved collaboration between content authors or improved access to shared content, unfortunately find themselves lower on the priority list. Thankfully, achieving a high level of information and content sharing within an organization is possible thanks to new XML technologies; with a little foresight and of course the proper tools, company executives everywhere are unlocking content and unleashing the hidden value stored away in company computers.

This paper focuses on two industry leading technologies, TEXTML Server and XML Spy 4 Suite, which, when combined, provide a powerful means of unlocking content. The technologies are both based on **XML** – e**X**tensible **M**arkup **L**anguage.

**IXIASOFT** and **Altova** are the leaders in their respective fields. To illustrate how easily these tools can help manage content, a basic web content publishing architecture will be presented as well as the description of a real-life application developed using this architecture.

**TEXTML Server** by IXIASOFT ([www.ixiasoft.com](http://www.ixiasoft.com)) is the leading native XML content server available on the market today. It is designed specifically to manage large volumes of semi-structured or unstructured content, providing a scalable content repository and powerful search engine.

**XML Spy 4 Suite** by Altova ([www.xmlspy.com](http://www.xmlspy.com)) is the leading XML authoring and development environment available today. It is designed to meet the needs of both developers and business users to create an effective framework for editing XML documents.

---

<sup>1</sup> Intellor Group Research Summary – XML Database Trends and Influences (2001)

# XML

---

XML – (eXtensible Markup Language) is a recommendation put forth by the World Wide Web Consortium (W3C) and is quickly becoming the dominant format for describing content on the Web. It was conceived to enable separation of content from presentation to make content more easily accessible across the Web. XML is a meta-language used to describe content and can be understood by virtually any software application. Using XML technologies, a business user could create content once, which could be published at any time, across many different mediums.

XML is designed to be both computer and human readable, and separates a document's content from its layout. The XML content can be presented in a user-friendly, professional format by means of using XSLT (XSL Transformation) which can transform an XML document into a format easily readable by people, or to any other format.

For example, an application can use various style sheets to present customized content according to the target audience. One style sheet for the web, another to publish the content to a PDA and another for print as shown below in Figure 1.

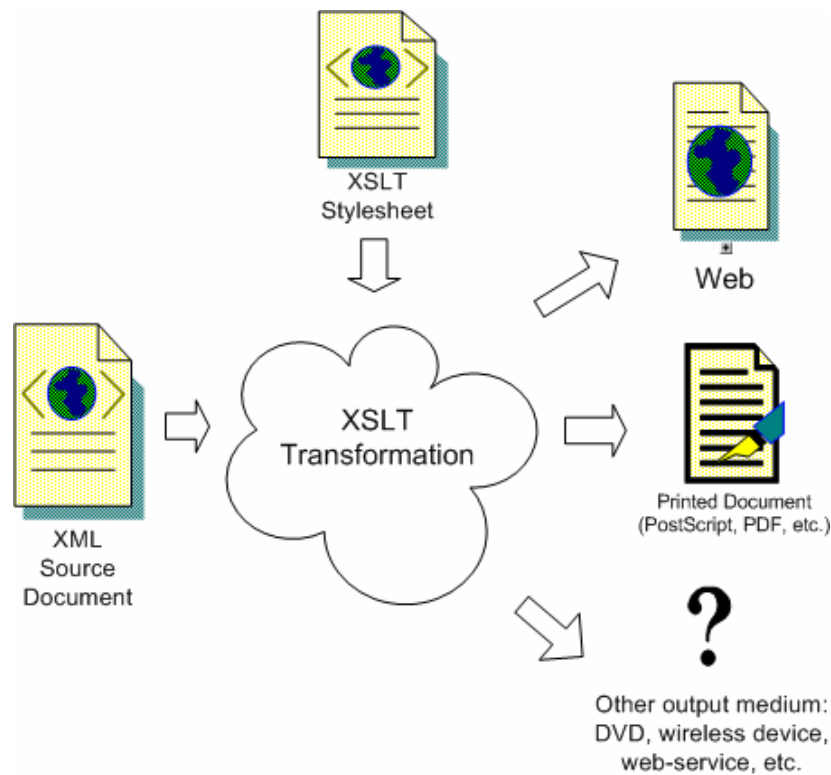


Figure 1: Publishing content stored in an XML document to multiple outputs by means of an XML Transformation.

As described above, TEXTML Server is a native XML content server. What exactly does this mean and what are its advantages over traditional relational databases such as SQL Server or Oracle?

### **What is a Native XML Content Server?**

A native XML content server stores XML documents in their original format, and is able to recognize the document's structure without mapping to a specific database model. The XML document is the fundamental unit of storage. As a result, the physical structure of the document is preserved 100% and is never stored in tables. In a native XML content server, there is no dependency on schemas or DTDs, and as such it is able to accommodate any well-formed XML document, making it ideal for managing semi-structured or unstructured content. Since there is no dependence on a predetermined structure, a native XML content server can store and retrieve heterogeneous document types and is well suited to applications where content varies in content and structure.

### **What is a Relational Database Management System (RDBMS)?**

A relational database is a collection of inter-related tables, each consisting of rows and columns. Data is stored inside these tables and all operations on data are performed on the tables themselves and produce additional tables as a result. When storing XML documents in a relational database, the various XML elements and attributes must be mapped to a pre-determined structure according to information contained in a DTD or XML schema, RDBMS are ideally suited to managing structured typed-data in tabular formats, especially if the data is not likely to change.

### **Data vs Document**

Why make a distinction between "data" and "document" driven applications? A mistake too often made is the one where we try to use familiar tools to solve new problems. As stated earlier, most organisations do not allocate resources to managing unstructured content. We can quickly conclude that developers have less experience in solving issues related to semi-structured or unstructured content. This often leads to developers storing loosely-structured documents within a column of a RDBMS table.

The application architecture is often based on the limitation placed by technology rather than the content. The determining factor in selecting the optimal content storage and management technology for your critical enterprise applications should in fact be the nature of content and how the application wishes to exploit that content. The first step is identifying the nature of your content.

### Is your content **data**?

- Highly structured information.
- The structure is not likely to be modified.
- Comprised of short fields of typed information, similar in length.
- Example: Financial data, Customer records, etc.

### Is your content **document**?

- Un-structured or semi-structured information.
- The structure is likely to change, and difficult to predict.
- Comprised of rich information, varying in length and type.
- The information may be a combination of text, pictures, voice and other multimedia components.
- Example: Legal briefs, contracts, technical documentation, patient health records, news articles, product descriptions, etc...

Once you have determined the type of information you will be managing, the next step is to select the most beneficial tool.

Let's briefly examine TEXTML Server's features:

- TEXTML Server is a repository, indexing engine and search engine whose functionality is exposed via API. It enables developers to create applications that will store and retrieve large volumes of content efficiently. Development with TEXTML Server is simplified by its clear, well documented COM+ and JAVA APIs.

- **Indexing**

- Index any well-formed XML document regardless of DTD or schema.
- Create context-specific indexes where the data type is specified:
  - Full Text
  - Numeric
  - String
  - Date/Time
- Modifications to the source or structure can easily be **updated on the fly** and do not require complex reprogramming of the database.
- Logically group related elements and attributes in specific, relevant indexes.
- Select only relevant information to be indexed, reducing database overhead.

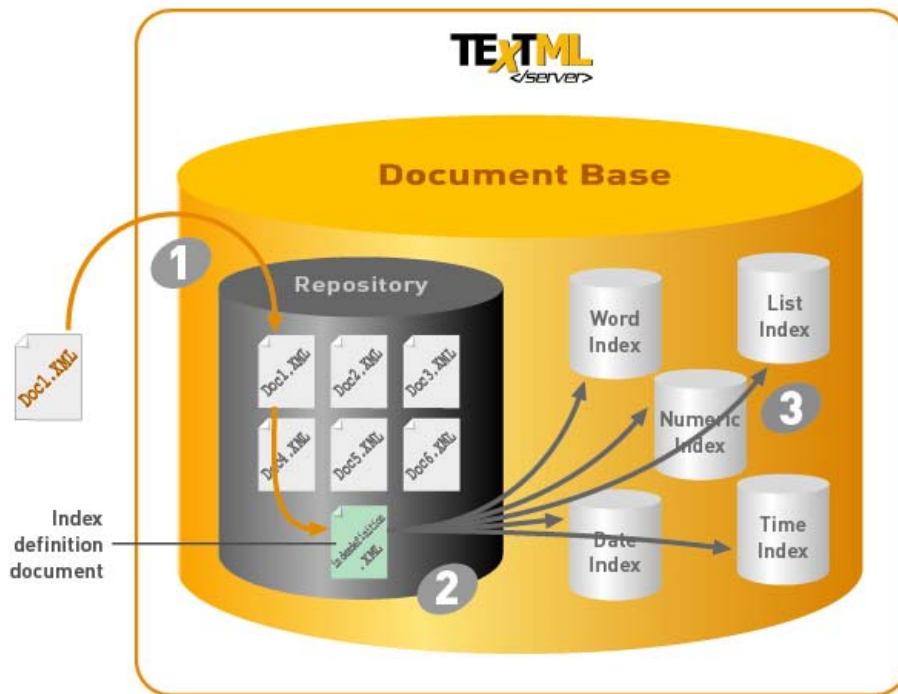


Figure 2: Repository and Index structure

- **Storage**

- Designed to store, index and retrieve millions of documents.
- The original document and all its content remain intact.
- The original document is never decomposed, or reconstructed such as with a relational database. This bi-directional transformation necessary in an application relying on a RDBMS is not negligible in terms of system resources.

- **Search performance**

- Simple yet powerful query language expressed as XML
- The only native XML content server specialized in full text searching
- Speed
- Flexibility - create searches that span multiple indexes of varying types. This allows you to create searches as complex as: Return all documents authored by John Smith containing the words "dog" and "cat" within 4 words of each other, created on May 20<sup>th</sup>, 2002.
- Accuracy
- Simplicity
- Can combine searches on the content and associated metadata of a document. Also combine searches with document properties.

## 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 for use within any XML repository or content management 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 business users across the company is immediately saved to an underlying XML format, ensuring that information is valid and does not become lost or un-usable. 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. 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.

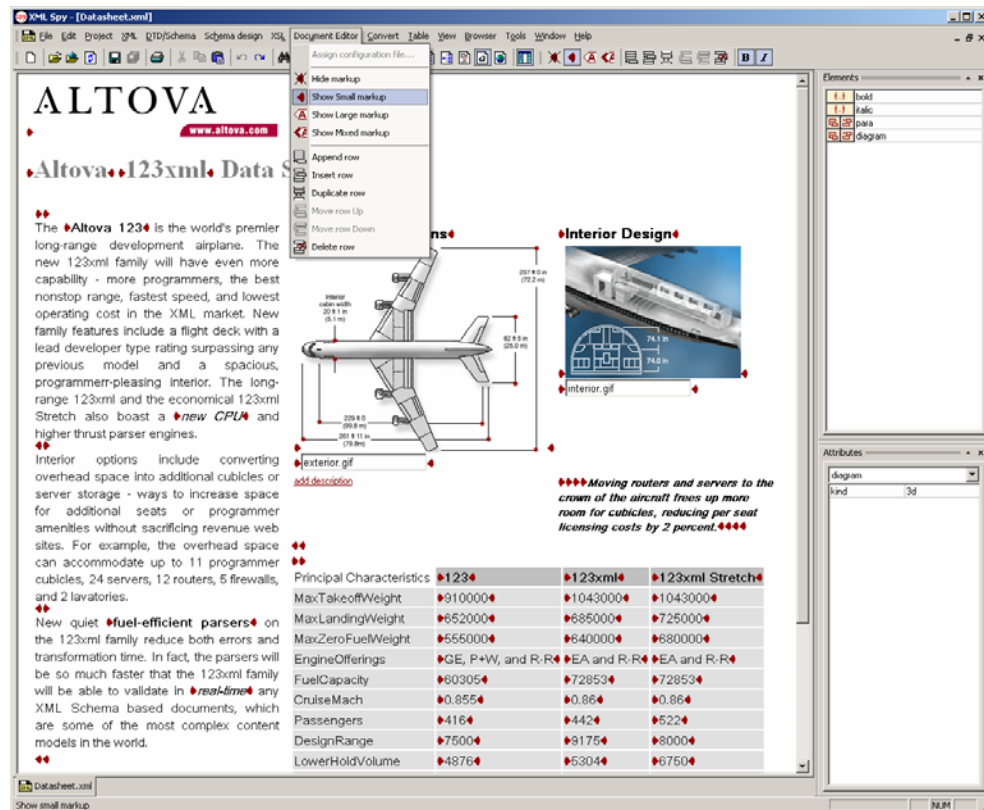


Figure 3: The XML Spy 4 Document Editor is shown above; it supports word processor like free-flow WYSIWYG text editing, form-based data input, graphical elements, presentation and editing of arbitrary repeating XML elements as tables, support for HTML and CALS Tables, real-time validation, spell-checking and consistency checking using XML Schema. The document template in the above screenshot was created using the XML Spy 4 XSLT Designer, which is a separate product that is included with the XML Spy 4 Suite.



The XML Spy 4 Document Editor is the only XML editor which can be deployed as a browser plug-in. In today's global economy, it is often the case that the business users who create content are located anywhere across the Internet, and as such, require easy Internet access to the underlying content storage systems. 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 web-based solution for rich-content editing currently offered in the industry.

## The XML Spy Integration kit

To enable a seamlessly integrated environment between TEXTML Server and XML Spy, IXIASOFT has created an Integration Kit. This kit can be installed on any machine where XML Spy and TEXTML Server (only minimum client installation is required) are installed. The kit then allows users of XML Spy to open and save documents to a TEXTML Server document base directly from the XML Spy environment, as displayed in figure 3 below. The user also always has the ability to save a document to the local file system should they choose to do so. The illustration below shows a user's view of the "File" command once the Integration kit is installed. In this case the user is choosing to "Add" a document to a TEXTML Server document base.

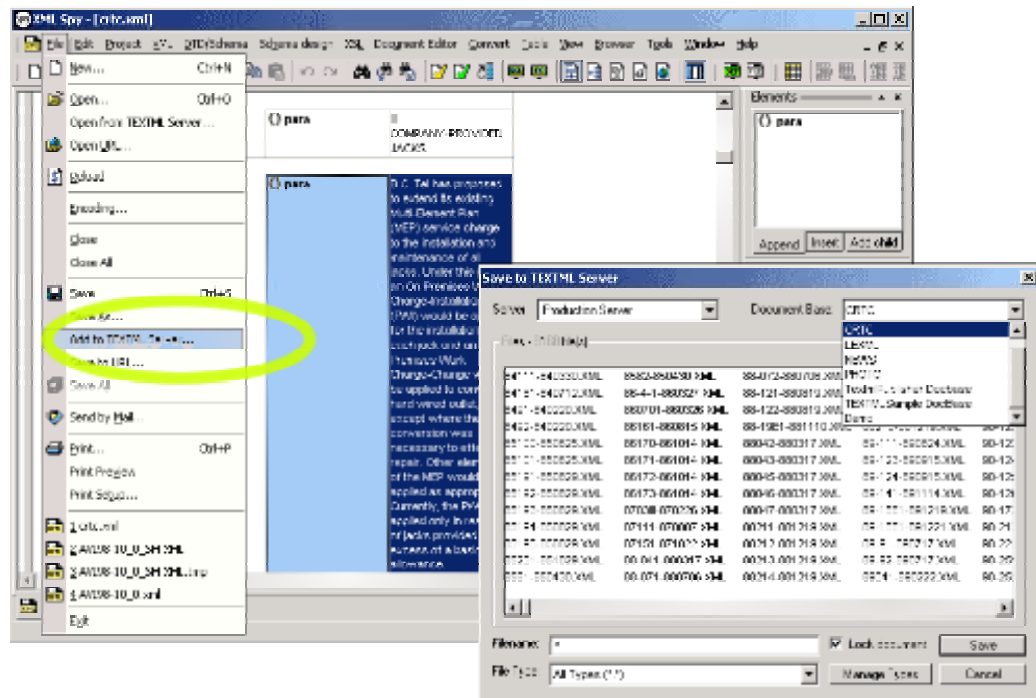


Figure 4: Opening XML documents stored in TEXTML Server directly from within the XML Spy Editing Environment.

Taking advantage of XML Spy's flexible 3<sup>rd</sup> party plug-in architecture, TEXTML Server API calls are available directly from within the XML Spy environment. The integration kit is a registered COM DLL written in C++. This strategy enables TEXTML Server to be directly accessible from the XML Spy menu.

By implementing TEXTML Server's document management features, administrators can:

- Set user privileges by document base to customize viewing and editing rights within applications.
- Check in and check out documents to prevent simultaneous document modifications.
- Lock a document base to enable "read only" document access while denying document base modifications.

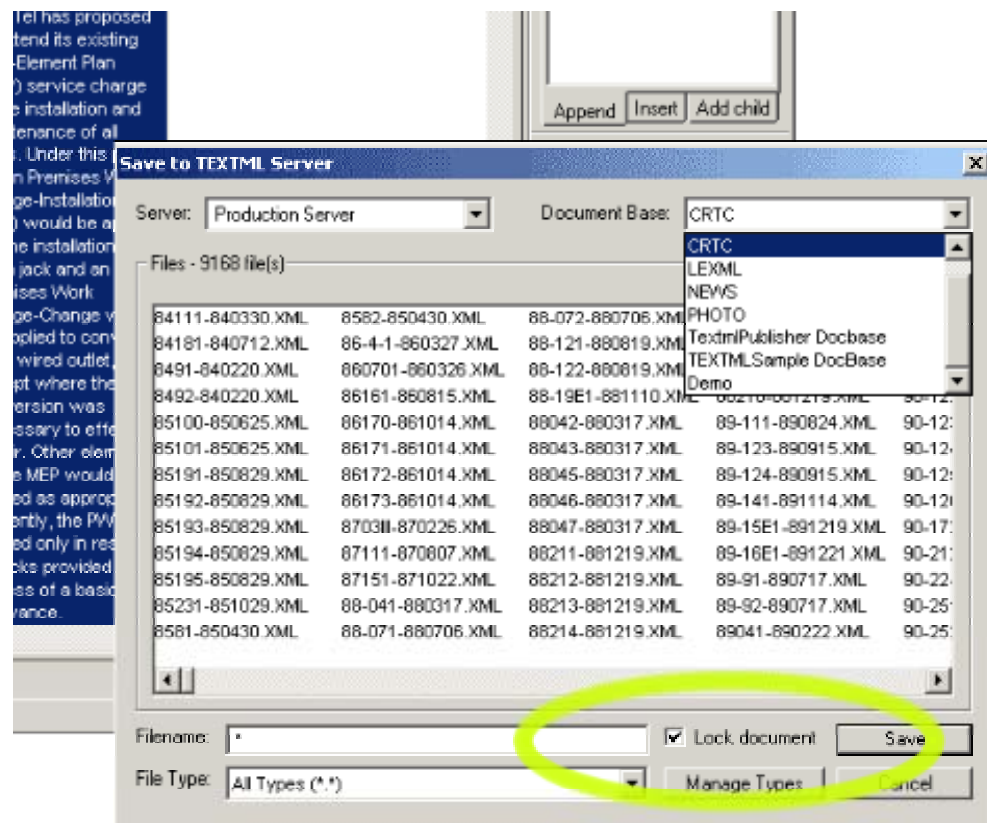
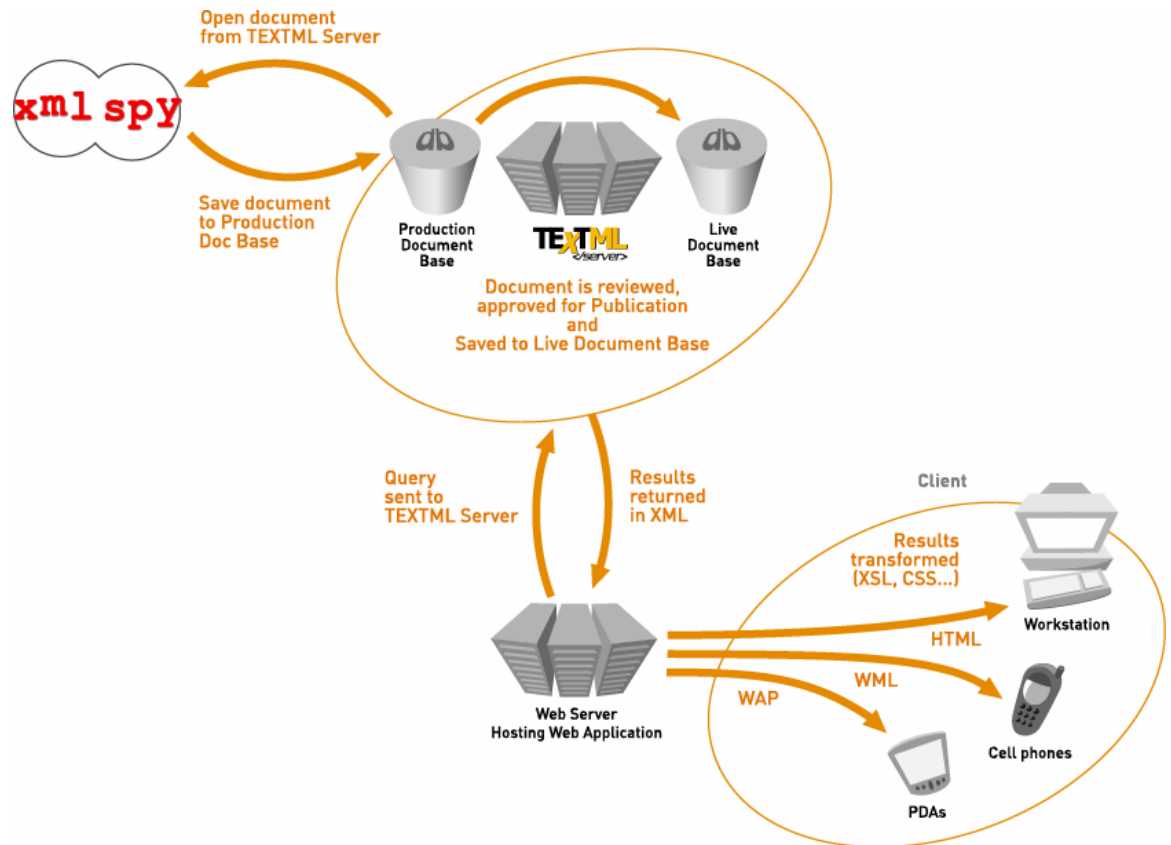


Figure 5: Locking documents as they are about to be edited

- Create valid XML by enforcing the use of a DTD or schema

# Sample System Architecture

Building a Web-publishing environment with TEXTML Server and the XML Spy browser plug-in.



In a web content publishing environment, there are basically three major steps: content creation, storage and publishing. The sample application we will be discussing is in fact a live demonstration of **TEXTML Server** and its integration with the **XML Spy browser plug-in**. To view this demo, simply go to <http://www.ixiasoft.com/onlinedemo/index.asp>.

TEXTML Server Online Demo - Microsoft Internet Explorer

New Search End Session

**TEXTML** Sample  
 </server>  
 Document editing provided by XML Spy

> Search for patent abstracts

Title:  Search Operators

Full Text:

Publication date - from: January 8 1920  
 to: March 28 2002

Patent numbers:  View List

Priority numbers:  View List

Application numbers:  View List

Assignees:

Inventors:

International patent classification:  View List

Sort by: Patent Number  
 Select 2nd criterion  
 Select 3rd criterion

Clear Search

This live demo enables you to search and modify 30,000 patent abstracts. Search and storage functionality provided by TEXTML Server and editing by XMLSPY. All documents are stored natively as XML and indexed by TEXTML Server. This demo allows you to search full-text, strings, numeric and date intervals and also offers the possibility of sorting search results by multiple criteria. Use the 'clear' button to remove previous search criteria.

**Priority Number**  
 Allows you to search for specific patents by "priority number" or browse the list of priority numbers found in the repository.

**Example:** Type "AM2000088 20000906" in the text box and click on "search". Or, alternatively, type "AM\*" and click "View List" to display the list of patents that start with "AM". You may then select a patent and click "Search" to display the patent abstract.

The demonstration consists of a searchable repository of 30,000 European patent abstracts, stored in XML format. Once an abstract is retrieved, the user has the ability to modify its content using the XML Spy browser plug-in, save the changes – which updates the indexes - and immediately retrieve the updated document. Of course in real life, one would not want to enable a patent to be modified over the web, however, this demo was created to showcase the dynamic indexing and search functionality of TEXTML Server, and to highlight the integration between TEXTML Server and XML Spy.

## Content Creation

---

Content fed to an XML web publishing application can be done in two basic ways – by converting the existing content from existing document formats to XML or by creating it directly in XML. The former will most likely be necessary to incorporate legacy data and the latter is becoming increasingly popular as the widespread use of XML editors such as XML Spy becomes more common. The level of sophistication and user-friendliness of XML editors has increased dramatically over the last few years and will continue to do so. For the purposes of this paper, we will not discuss legacy data conversion.

TEXTML Server Online Demo - Microsoft Internet Explorer

New Search End Session

TEXTML Sample /server>

Document editing provided by XML Spy

View TEXTML Document Properties View Hit Info Edit with XML Spy

### NOVEL AMIDE COMPOUNDS AND DRUGS CONTAINING THE SAME

Patent number	Equivalent(s)
AU1053899	CA2303781 HU0000729 WO9925712

Publication date : 2000-02-16  
 Priority Number(s) : WO1998JP05149 19981116  
 JP19970330877 19971114  
 Application number : EP19980953063 19981116

Assignees	Inventors
KOWA CO (JP)	KITAMURA TAKAHIRO (JP) OZAKI CHIYOKA (JP) SATO YUKIHIRO (JP) EDANO TOSHIYUKI (JP) MIURA TORU (JP) OHGIYA TADAAKI (JP) HIRATA MITSUTERU (JP) KAWAMINE KATSUMI (JP) SHIBUYA KIMIYUKI (JP)

European patent classification	International patent classification
<a href="#">C07D417/12+27B+221B</a>	C07D403/12
<a href="#">C07D401/04+263B+221B</a>	C07D413/12
<a href="#">C07D401/12+235C+221B</a>	C07D417/12
<a href="#">C07D413/12+263B+221B</a>	A61K31/44
<a href="#">C07D413/12+263B+239B</a>	A61K31/505

The present invention provides to a novel compound having an ACAT inhibiting activity. The present invention relates to compounds represented by formula (I) wherein represents an optionally substituted divalent residue such as benzene, pyridine, cyclohexane or naphthalene, or a group, Het represents a 5- to 8-membered, substituted or unsubstituted

In this demonstration, we have chosen to incorporate the **XML Spy browser plug-in** that allows a client using Microsoft Internet Explorer 5.5 or above to directly edit XML documents in IE. The 3<sup>rd</sup> party plug-in architecture of XML Spy enabled IXIASOFT developers to create an integrated bridge between the two technologies. The COM API of XML Spy only required two simple ASP pages – one to check out a document from TEXTML Server and one to check it back in.

Once a document is checked out, the browser editor takes over. At this point the user is in a document editor environment right in their own browser. The browser plug-in is completely customizable and there are any number of operations which can be performed on the document. Customization is done via simple client-side scripting. Insert a new element or attribute, validate the document, add customized formatting, printing, table modifications and so on. The simplicity of this integration is its real strength. In addition, every document exchanged between XML Spy and TEXTML Server is an XML document. TEXTML Server stores XML documents natively and there is no internal decomposition of the XML document; with a RDBMS, a bi-directional XML transformation would be required and would inevitably reduce performance significantly as the application scales.

## Storage

---

TEXTML Server is designed to efficiently manage millions of XML documents and as such is ideally suited for large-scale enterprise-wide applications. TEXTML Server is able to store any binary file format and can index any well-formed XML document. TEXTML Server also enables document level locking as well as the ability to lock an entire document base.

As content providers create new content using XML Spy, they can save documents directly into a TEXTML Server document base located locally or somewhere on their network. This facilitates collaboration between multiple authors who contribute content directly to a centralized, searchable repository from any location.

When an author selects **/File/Open from TEXTML Server/...** from the XML Spy Menu bar, they are actually connecting to TEXTML Server, selecting a document base, searching for document, performing a 'get document' and locking the document all from the XML Spy environment.

## Publishing

---

This final step is the most crucial as it is the closest to the end user, the consumer of the published content. TEXTML Server is the ideal solution for publishing XML content over the web. This is accomplished by a simple yet powerful indexing strategy coupled with TEXTML Server's powerful search engine.

The following is a description of the flow of information as it occurs in the Patent Abstract Demo from the moment a user logs in to the application. There are several steps so it is recommended to login to the demo site and follow along... <http://www.ixiasoft.com/onlinedemo/index.asp> .

*To view a sample document, simply login to the sample, perform a search and select "View Native XML" on any of the documents presented in the results set page.*

### Login

As a user logs in to the demo application...

- 1- The user connects to TEXTML Server
- 2- A document base (Patents) is selected

### Dynamic search interface generation

- 1- A query is sent to TEXTML Server (Retrieve 'indexdefinition.xml')
  - a. The index definition<sup>2</sup> is a set of rules expressed as an XML document governing how TEXTML Server will index the content in a particular document base.
- 2- The index definition is then processed to generate a search interface.
  - a. Each text box title (*Full Text, Patent Numbers, Priority Numbers, etc..*) represents a separate index that is

---

<sup>2</sup> For more information on index definitions in TEXTML Server, please refer to the IXIASOFT documentation center - [http://www.ixiasoft.com/support/doc/textml\\_server\\_doc.asp](http://www.ixiasoft.com/support/doc/textml_server_doc.asp)

searchable. Each one of these indexes is represented in the index definition.

### Query<sup>3</sup> Generation

- 1- The user types their search criteria (for example 'Television' in the 'Full Text' index) and hits enter.
  - a. This action dynamically creates a query performed on the indexes of the document base.
    - i. Search criteria is compared to the index definition, and a query generated, also expressed as XML.
    - ii. In the case of a query that searches for 'Television' in the "Full Text" index, the query is expressed as:

```
<?xml version="1.0" ?>
<query VERSION="2.0" RESULTSPACE="R1">
  <key NAME="FullText">
    <elem>television</elem>
  </key>
</query>
```

*To view any query generated by the sample, perform a search and click on "View Query" in the results set page.*

### Result Set

- 1- The query is sent to the server, the indexes are searched and a result set is returned
  - a. A result set consists of an object containing pointers to all the documents stored in the server along with their associated document properties.
- 2- Two frames are created to present the results
  - a. The left frame will display x-number of documents and the right frame is left available until the user selects a document for viewing
  - b. The result set is analysed and loaded into a DOM
  - c. The document name and patent title are extracted and displayed in the left hand frame.
- 3- Once the user selects a document, it is displayed as HTML in the right hand frame. The user can always view the native XML file by selecting the 'view native XML' button.
  - a. At this point, the user can perform several operation on the server:
    - i. View TEXTML Document Properties
    - ii. View Hit info
    - iii. View Query
    - iv. Sort results
    - v. Edit with XML Spy

---

<sup>3</sup> For more information of TEXTML Server's XML Query language, please go to [http://www.ixiasoft.com/support/doc/textml\\_server\\_doc.asp](http://www.ixiasoft.com/support/doc/textml_server_doc.asp)

- 4- If the user selects to 'Edit with XML Spy',
  - a. A 'getdocument' is performed to retrieve the document
  - b. The document is locked to prevent others from modifying the same documents simultaneously.
    - i. The document is now in read-only mode and continues to be available for search even while another user modifies the document.
  - c. Modifications are made and the user clicks "save".
    - i. A "setdocument" is performed.
    - ii. The document is flagged for indexing
    - iii. The indexes are updated
  - d. The user then closes the document.
    - i. The document is unlocked, making it fully read-write accessible to all users.
  - e. The user may then search for the same document by searching for the updated information.

## Conclusion

---

XML has quickly become the dominant Web content format and delivers the ability to unlock knowledge hidden throughout your company. The combination of TEXTML Server and the XML Spy 4 Suite will provide you with a scalable, high performance XML content authoring, repository, search and publishing technology that will streamline the XML delivery process and enable you to take advantage of content.

Needs analysis is the first step towards effective web content publishing and is often the hardest. This paper has introduced you to two technologies that offer an integrated XML Authoring and Publishing environment, making this first step an easy one.

**TEXTML Server Evaluation Edition available here**

<http://www.ixiasoft.com/textmlserver>

**XML Spy downloads**

<http://www.xmlspy.com/download.html>

**TEXTML Server XML Spy Integration Kit**

<http://www.ixiasoft.com/xmlspy>