

## Native XML Databases – A Review

*by Vijay Murthi*

Project	Developer	License	DB Type	Pros	Cons
<a href="#"><u>4Suite</u></a> , <a href="#"><u>4Suite</u></a> <a href="#"><u>Server</u></a>	Four Thought	Open Source	Object-oriented	<ul style="list-style-type: none"> <li>• Command Line Interface</li> <li>• APIs in Python, HTTP, SOAP and XSLT.</li> <li>• Supports remote, cross-platform and cross-language access through HTTP, RPC FTP and CORBA.</li> <li>• Java API's</li> <li>• Last release on September 30, 2002</li> </ul>	Installation requires both python & 4 suite software.
<a href="#"><u>eXist</u></a>	Wolfgang Meier	Open Source	Relational	<ul style="list-style-type: none"> <li>• Appears to be a strongly developed project with sufficient docs.</li> <li>• It may either run as a standalone server process, embedded into an application or in connection with a servlet engine.</li> <li>• Last release on January 14, 2003</li> </ul>	<ul style="list-style-type: none"> <li>• Relational database.</li> <li>• No mention about command line support.</li> </ul>
<a href="#"><u>DBDOM</u></a>	K. Ari Krupnikov	Open Source	Relational	[note]	<ul style="list-style-type: none"> <li>• An implementation of DOM in SQL, on a relational database platform.</li> <li>• Turns an RDBMS into an XML application server.</li> </ul>
<a href="#"><u>Lore</u></a>	Stanford University	Research	Semi-structured	[note]	The Lore project was declared a success in the year 2000 and is now pretty much out of business.

[note] – The Cons completely outweigh any other advantage the software might have to enable us to use with JIM.

<u>ozone</u>	ozone-db.org	Open Source	Object-oriented	[note]	<ul style="list-style-type: none"> <li>• Complete Java &amp; enterprise stuff</li> <li>• Last release by July 29, 2002</li> <li>• No command line interface.</li> </ul>
<u>XDBM</u>	Matthew Parry, Paul Sokolovsky	Open Source	Proprietary (Model-based*)	[note]	<ul style="list-style-type: none"> <li>• Last updated by August 23, 2000</li> <li>• Most of the links not found</li> </ul>
<u>Xindice</u>	Apache Software Foundation	Open Source	Proprietary (Model-based*)	<ul style="list-style-type: none"> <li>• Xindice supports three APIs: the <u>XML:DB API</u> (also from the XML:DB Initiative), a CORBA API, and an XML-RPC plugin which supports access from languages such as PHP, Perl, and Applescript.</li> <li>• Command Line Interface</li> <li>• XQuery support on high priority TO-Dos plus already support XPath. [XQuery and XPath are the two popular query languages for XML databases]</li> <li>• Ease of use &amp; flexibility.</li> </ul>	Need Java for Server installation.

*Native XML databases fall into two broad categories:*

\*Model-based **storage** Store a binary model of the document (such as the DOM or a variant thereof) in an existing or custom data store

**Text-based storage** Store the entire document in text form and provide some sort of database functionality in accessing the document.

*There are two major differences between the two strategies:*

- First, text-based storage can exactly round-trip the document, down to such trivialities as whether single or double quotes surround attribute values. Model-based storage can only round-trip documents at the level of the underlying document model. This should be adequate for most applications but applications with special needs in this area should check to see exactly what the model supports.
- The second major difference is speed. Text-based storage obviously has the advantage in returning entire documents or fragments in text form. Model-based storage probably has the advantage in combining fragments from different documents, although this does depend on factors such as document size, parsing speed (for text-based storage), and retrieval speed (for model-based storage).