

TEST CENTER EXCLUSIVE
Oblix ShareID 2.0
Federation Server p24

**Siebel Looks
Beyond CRM** p16

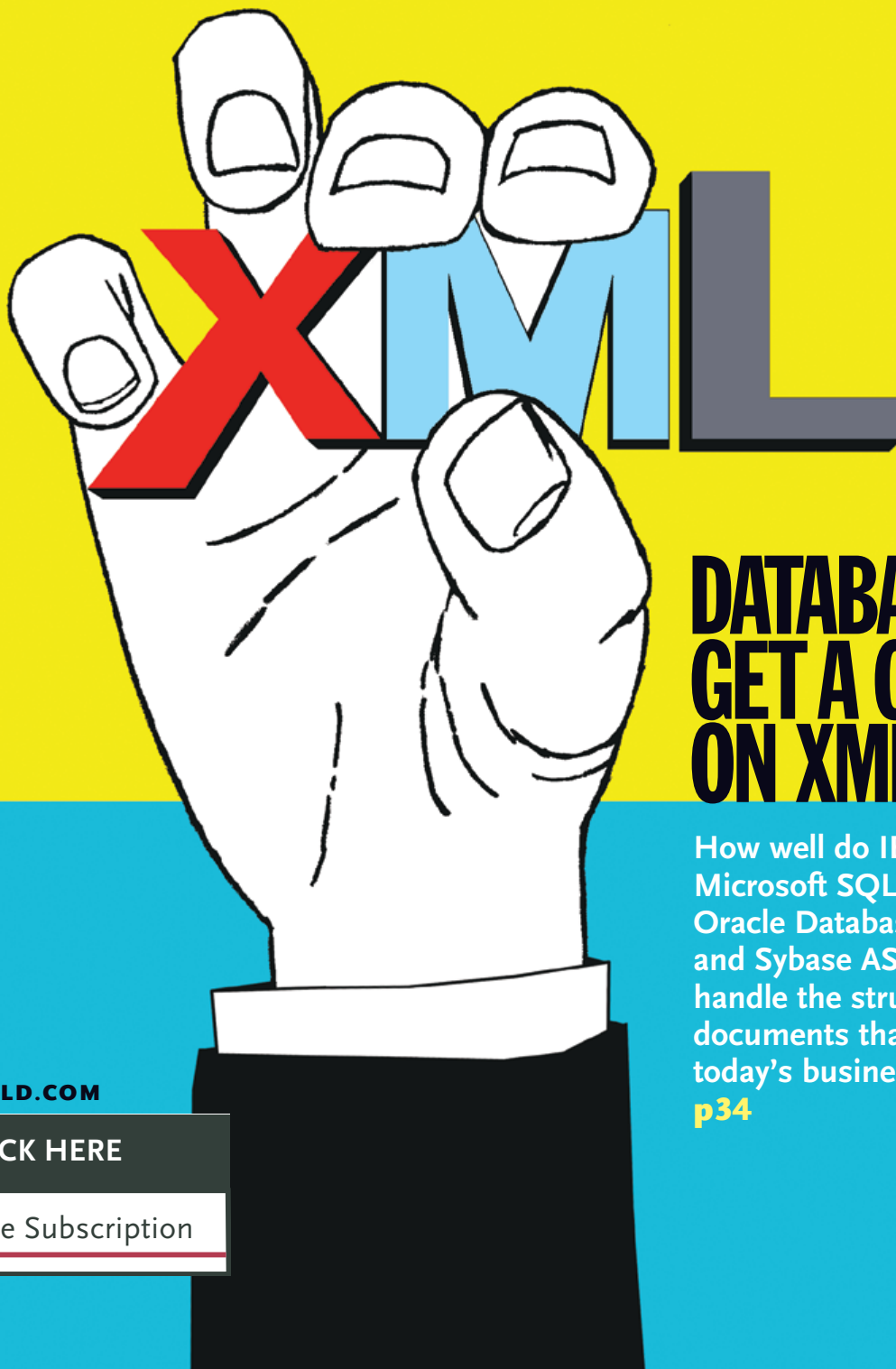
JON UDELL
**Seeking Better
E-mail Clients** p32



InfoWorld®

April 26, 2004 ■ Issue 17

GET TECHNOLOGY RIGHT™



DATABASES GET A GRIP ON XML

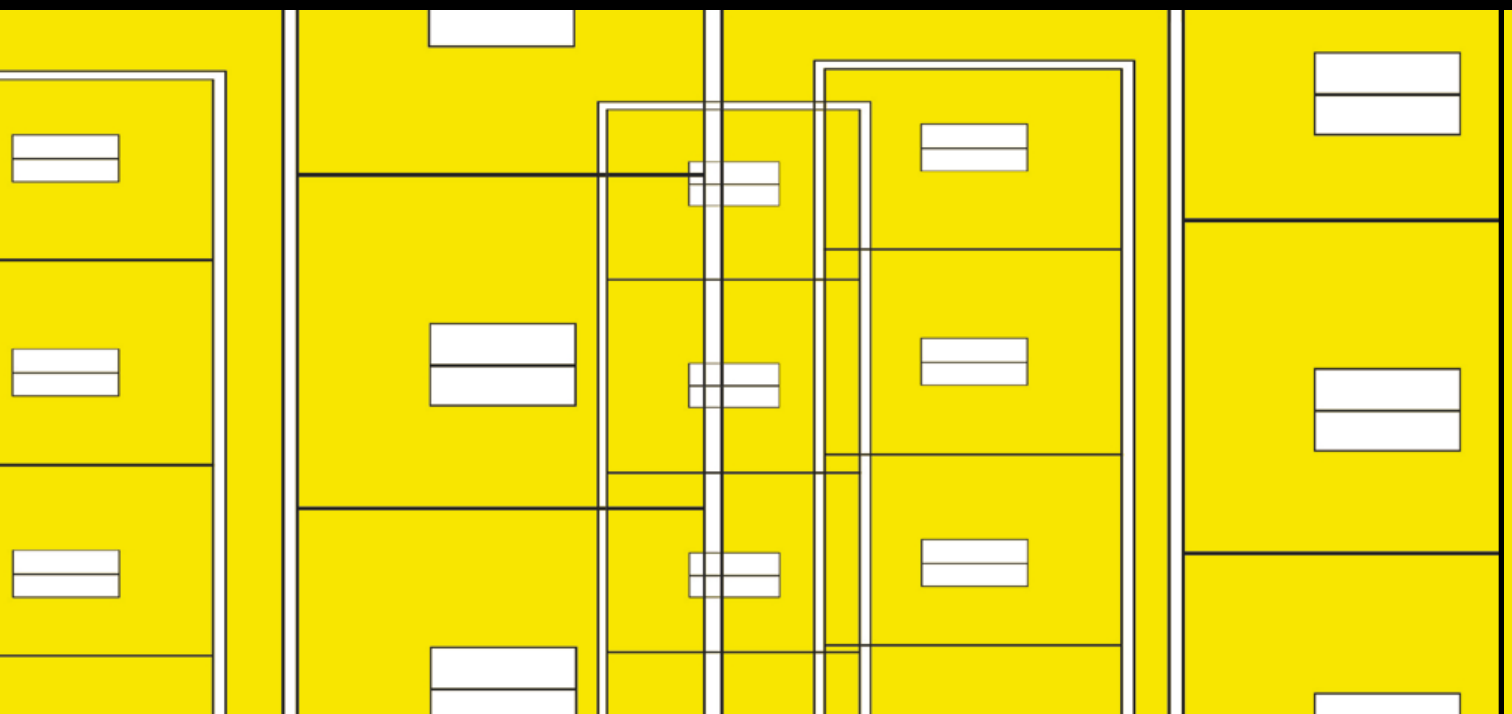
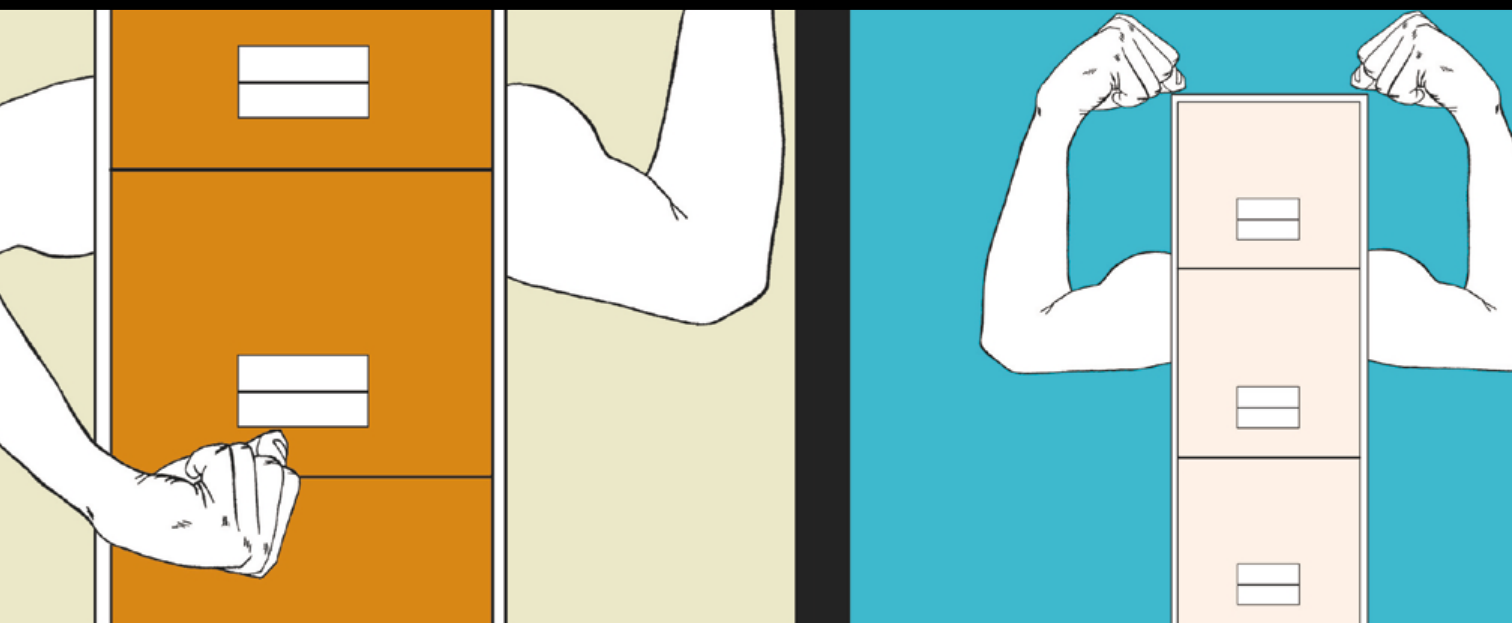
How well do IBM DB2, Microsoft SQL Server, Oracle Database 10g, and Sybase ASE handle the structured documents that drive today's business?

p34

 **INFOWORLD.COM**

 **CLICK HERE**

For a Free Subscription



IBM, Microsoft, Oracle, and
Sybase compete in our
data-management gymnastics

Databases Flex Their XML

IF YOU COULD DO ONE THING TO IMPROVE INTEGRATION AND automate processes with customers and business partners, it would be to implement XML, which has become the standard for exchanging information between disparate systems because it is easily transformed into any format. With very little effort, the same file can be sent to several different customers with their own specific needs. XML eases the development effort for the transmitting company and gives recipients a safety net for altering the way they use the data without having to alter how they receive it.

BY SEAN McCOWN | ILLUSTRATION BY MARK DANIELSON

Perhaps the most important improvement in Oracle Database 10g is support for schema evolution.

Being able to merge, query, and transform transmitted data with relational data is becoming as essential to businesses as data warehouses themselves. The good news is that the four leading relational databases, namely Oracle Database, IBM DB2, Sybase ASE (Adaptive Server Enterprise), and Microsoft SQL Server, not only can store XML data, but they hide much of the complexity of working with XML. Depending on which of these relational databases you use, however, the XML features you will have to work with may be extremely rich or limited in important ways.

What does a fashionable XML database provide? Four basic functions: the

ability to consume, store, search, and generate XML. The extent to which the database supports these functions and the methods it uses to accomplish them are what make for a successful implementation of XML in a database.

I examined these four areas in Oracle Database 10g, IBM DB2 Universal Database V8.1, Sybase ASE 12.5.1, and Microsoft SQL Server 2000. I tested how they imported and read XML files, their options for saving the data, their indexing and query capabilities, and their options for creating XML and graded them based on the ease, flexibility, and speed with which they handled the most common XML operations.

Of course, these products have many other capabilities beyond handling XML. My grades should not be interpreted as complete evaluations.

Documents and Tables

Relational databases and XML documents are both powerful ways to represent relationships among data, but they're powerful in different ways. For example, querying on a patient ID number in a relational database may allow you to quickly find the dates a certain patient visited the hospital, the conditions he was diagnosed with, and the treatments he was given. But it likely won't help you determine which treatments were provided for which

SCORECARD

Which Relational Databases Make the XML Grade?

Oracle has the richest, smoothest XML handling capabilities; IBM and Sybase are solid performers; and Microsoft has some catching up to do.

	Consume	Store	Search	Generate
IBM DB2 Universal Database 8.1	Importing XML files into the database was intuitive and fast	Stored XML using all three methods, validated schemas, and fully indexed the data	Easily created views combining XML and relational data but lacked XQuery support	Generated XML files faster than the rest, complete with XSLT transformations
Overall grade B	Grade A	Grade A	Grade B	Grade A
Microsoft SQL Server 2000	Offered many options for pulling out specific data	Inability to store structured XML is a serious deficiency; XML data could be shredded into the database but not validated	Imported XML data is converted to relational data, so granular XML search capabilities are lost	Provided a full set of features to create XML files, but working with the .Net Framework's XML provider is often easier
Overall grade C	Grade A	Grade C	Grade C	Grade B
Oracle Database 10g Release 1	Provided rich methods for consuming XML, including files delivered via HTTP and FTP	Supported all three methods of XML storage plus interaction with WebDAV repositories	Provided the easiest, most comprehensive searching capabilities	Generated XML files quickly and easily; breezed through complex transformations
Overall grade A	Grade A	Grade A	Grade A	Grade A
Sybase ASE 12.5.1	Outstanding job of consuming XML documents	Supported all three flavors of XML storage and provided rich indexing capabilities	Offered many options for creating views and encapsulating queries but no XQuery support	Generated XML files very fast but lacked on-the-fly XSLT transformations
Overall grade B	Grade A	Grade A	Grade B	Grade C

conditions or what times the treatments took place, nor will it give you other useful information that XML versions of these records could provide.

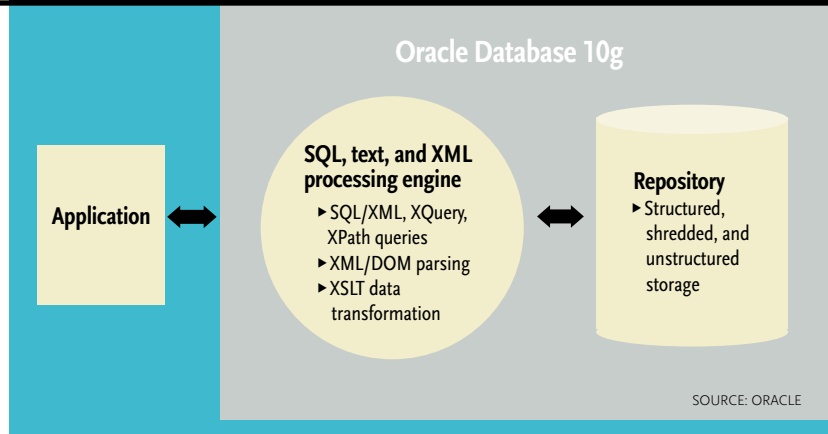
But whether or not you can combine the benefits of relational and XML data depends on how you store the XML. There are three methods for physically storing XML data in a relational database: shredded, unstructured, and structured. Shredded and unstructured are useful methods but limited. The structured method allows you to leverage the power of both relational data and XML hierarchies.

Shredding puts XML data into relational columns but strips it of its XML-ness, meaning the hierarchical relationships among the data in the original XML document are lost. Shredding is useful when you're not concerned about keeping the data in XML format. For example, let's say you have a Web site that allows customers to place orders, and the order needs to go to a number of different database systems. Producing an XML file and having the different systems pick it up — that is, shred it — from a network share may be the most efficient and error-free way to get the data where you want it to go.

The unstructured method uses a data type called a CLOB (Character Large Object) to store an entire XML document as a single unit. Databases have been doing this for years with different types of documents, so this is nothing new. The unstructured method provides limited search capabilities, but it is still quite useful. You can't base queries on it, but the structure of the original data is preserved. A good use for unstructured XML storage would be in keeping original documents to comply with government regulations. For example, if a financial institution were to receive original loan documents

How Oracle Gets It Done

Oracle Database processes SQL, text, and XML in the same engine, allowing for a high level of data integration and lightning-fast queries.



in XML, this would allow them to have a relational record of each loan application, and also to store the original application with that record.

The structured method allows you to store XML data inside the database and preserve the hierarchy of the data. Structured storage, also known as “native XML” storage, is what every vendor is trying to achieve. The most obvious benefit of preserving the hierarchical relationships of XML data is being able to receive an XML document, combine it or manipulate it with relational data, and produce XML as a result. It isn't possible to produce such result sets with a relational query language alone.

Only Microsoft SQL Server 2000, among the four databases tested, does not support structured XML storage. And while all four, including Microsoft, allow you to reap the benefits of the shredded and unstructured methods, the approaches they take and flexibility they provide can differ considerably.

Oracle Sets the Curve

Oracle Database 10g breaks new ground in support for XML technology, offering very rich features for importing, storing, querying, and generating XML data. Providing native, structured XML stor-

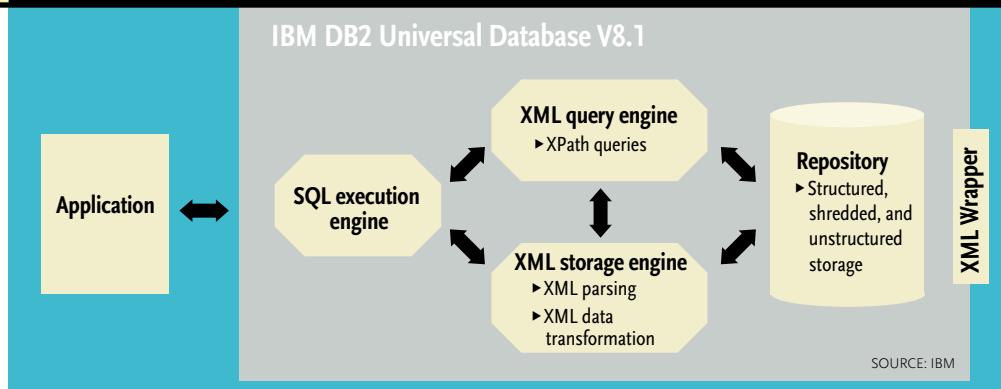
age as well as support for unstructured document storage and shredding, Oracle Database 10g allows you to pull XML data from files and merge it with relational data in views. But before jumping into an upgrade for the enhanced XML capabilities, Oracle shops should note that most of the functionality is available in Database 9i.

Oracle Database can read and write to WebDAV (Web Distributed Authoring and Versioning) repositories, giving users a nice Web folder view of the XML data they can request. The WebDAV support allows administrators to set up access to thousands upon thousands of XML files without requiring much disk space. This is because these WebDAV files, though they appear to be regular XML files, are just shortcuts, queries that point back to the database. The data that comprise these files doesn't materialize until it's requested by opening the file or copying it to a different location. Along with WebDAV, Oracle also supports transferring XML documents across FTP and HTTP.

Perhaps the most important improvement in Database 10g is XML schema evolution. Schema evolution allows you to implement changes to XML schema by mapping the existing data to the new

How IBM Gets It Done

DB2's XML functionality is extended by the Information Integrator's XML Wrapper. Located on the file system, the XML Wrapper exposes the database to potential performance and security issues.



them, so there are some very advanced possibilities here.

How easy is managing XML data in Oracle? Assuming you already understand all things XML, including SQL/XML, XQuery, and DTD (Document Type Definition), XML processes in Oracle Database are very easy

schema. Instead of having to export and re-import all of your XML data, you just create the XSLT (XSL Transformation) style sheet to transform your old documents to the new schema, and the database takes care of the rest. Schema evolution greatly eases the management of XML data, because one thing administrators can count on is constantly changing requirements.

Oracle Database 10g offers some superb options for storing XML data. The XMLType data type obscures much of the complexity of storing and querying XML data, allowing admins to manipulate the data using familiar

SQL tools and concepts. For instance, you can create an XMLType table or an XMLType column in a relational table. Creating an XMLType table is how you implement the structured storage method and preserve your XML hierarchy. Creating an XMLType column in a relational table is how you implement the unstructured storage method, which is useful for storing entire XML documents for attachment to relational records.

And what can be done with a table can be done with a view. You can also use these tables and views in stored procedures and process data based on

to manage. In fact, as in schema evolution, the database handles many data management processes on its own, with a minimum of admin intervention.

In my speed tests, which involved importing thousands of files, creating thousands of files, and importing very large documents, Database 10g proved very fast, beating all comers at imports and finishing a close second to IBM in the file-creation test.

Big Blue, B Student

IBM is on the ball, too. DB2 implements XML as a collection of options called the XML Extender. Another key

XML Essentials Oracle leads the pack with an impressive set of XML handling capabilities. IBM provides the essentials, Sybase lacks a few key pieces, and Microsoft just isn't ready for prime time.

	Supports XML shredding	Stores structured XML	Validates schema	Interacts natively with stylesheets	Merges XML with relational data	Supports XQuery	Supports SQL/XML	Supports Web services	Reads/writes to WebDAV repositories
IBM DB2 Universal Database 8.1	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	No
Microsoft SQL Server 2000	Yes	No	No	Yes	No	No	Yes	Yes	No
Oracle Database 10g Release 1	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Sybase ASE 12.5.1	Yes	No	Yes	No	Yes	No	Yes	Yes	No

piece, the Information Integrator, is licensed separately from DB2. Although the XML Extender may sound like a separate program that sits outside the database and interacts with the relational engine, it is in fact just a collective name for the stored procedures and functions that provide the XML functionality inside DB2. These stored procedures and functions sit inside the relational engine like any other objects in the database and are accessible to any user who needs them. It's only slightly annoying that there's a separate install to get the Information Integrator functionality.

The Information Integrator sits outside the relational engine and extends the functionality of XML in DB2 by adding the XML Wrapper, which allows you to treat XML documents as relational sources. The XML Wrapper is a very powerful tool that could be useful in receiving transmitted XML files that you want to be able to query and possibly report on, but which you don't want to import into the database.

Like Oracle Database, DB2 allows you to store structured XML, unstructured documents, or shredded data. And like Oracle's XMLType, DB2's XML datatype allows you to obscure the details of XML data storage from both users and admins alike. Views can be created to merge relational data with XML data in a single result set. This same result set can be merged with XML documents on the file system or with other XML or relational data inside the database.

DB2 uses SQL/XML as the main language for creating XML documents from relational data; you use SQL/XML functions to create XML tags, attributes, and the like. SQL/XML also allows you to perform concatenations and aggregations while creating the document.

XML Database Futures

AS SQL AND XML CAPABILITIES converge in the leading relational engines, you might conclude that there's no future for dedicated XML databases. A year ago, in his keynote at the InfoWorld CTO Forum, BEA Chief Architect Adam Bosworth explained why they're still relevant, and what their new mission will be. Coarse-grained XML messages are the wefts of the service-oriented fabric we're now weaving. We're going to be pumping a lot of those messages, and the pumps won't merely route messages. They'll also scan SOAP headers, correlate messages, enforce security policies, and respond to XPath-style queries — all this in a real-time environment that can't tolerate the slightest delay.

Oracle Database, IBM DB2, Sybase ASE, and Microsoft SQL Server were built from the ground up to satisfy a different set of requirements: read-write multiuser access to transactional data stores governed by well-defined schemas. Message brokers, by contrast, live in a world of transient messages that aren't written by multiple concurrent users, aren't for the most part subject to the ACID (atomic, consistent, durable, isolated) transaction model, and aren't always fully schematized. Sure, you could use an XML-savvy relational engine to power an XML message pump. But that would be like driving a Hummer to work: a big, heavy, expensive vehicle that was made for a different purpose.

If you've already bought your SQL/XML Hummer, though, it's rea-

sonable to want to make the most of that investment. Do you really need to buy an XML sports car, too, and become acquainted with a whole new set of quirks and foibles? The answer most likely is no. What you probably will buy, at some point, is an enterprise service bus (such as Sonic ESB) or a services fabric (such as webMethods Fabric). It will embed an XML database that may once have been a stand-alone product. For example, Sonic XML Server — once known as eXcelon — is vanishing into the woodwork of Sonic ESB.

Will Software AG's Tamino wind up similarly embedded? Bill Ruh, the company's CTO, isn't making any such announcement, but he doesn't discount the possibility either. "We're well suited to create an XML-based enterprise service bus," he says. "Our mediator can set rules around XML data, do message fan-in and fan-out, call other apps, wait, and make decisions based on the values returned."

It's true that you can use native XML databases to manage the growing number of business documents created by the new generation of XML-savvy end-user applications. It's handy, for example, to search an insurance database for incident reports that match some structured pattern of in-line metadata. But hybrid SQL/XML databases can do that too, and they can also join the structured XML content with relational columns — a powerful combination. So XML databases are migrating into a niche that SQL/XML can't and won't occupy. They're becoming the high-performance pumps that push XML traffic around on the emerging services web.

— Jon Udell

Dawn of the ESB

An enterprise service bus serves as an XML message broker.

- Routes XML documents based on message headers and contents
- Performs XML processing, queries, transformations, and storage
- Provides authentication, authorization, and encryption, and enforces security policies across the bus
- Provides business process management and monitoring capabilities

How Sybase Gets It Done

Sybase ASE uses separate engines for SQL and XML functions; this means separate calls are required for XML queries but have little impact on performance.

DB2's SQL/XML support isn't complete, but it covers the major functions. If you are using DB2 today, you'll probably be able to do anything you want when creating XML documents.

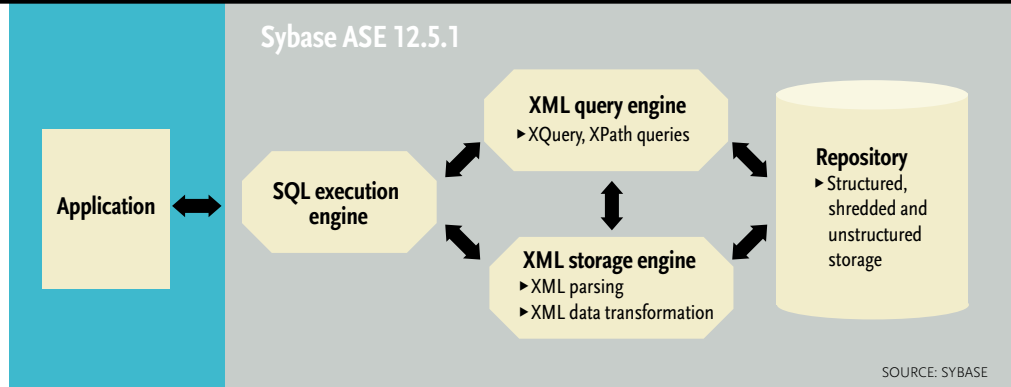
XQuery, on the other hand, is completely unsupported by DB2.

No doubt IBM is waiting for the W3C to solidify the specification before it implements the unfinished standard. Nevertheless, the absence of XQuery imposes significant limitations on DB2's XML search capabilities. I would much prefer to see a partial implementation than for it to be completely left out.

XQuery, which is a language used for crawling through structured data paths, offers significant query functionality beyond what SQL can do. Because XQuery is an identity-based query language, whereas SQL is value-based, you can use XQuery to find out whether a certain element exists; SQL can tell you whether an element that's known to exist holds a certain value. Unlike SQL, XQuery also has knowledge of attributes inside element tags, and it has stronger data types than SQL, providing you greater control over the type of data being written or queried.

XQuery notwithstanding, IBM's commitment to XML is quite apparent when you look at the improvements it has made in DB2 V8.1. Not only does V8.1 support XSLT for on-the-fly translation of XML data, there are more than 1,000 changes to SQL syntax for handling XML alone.

How easy is DB2's XML to manage? Very. The views manage the dynamic data underneath them and provide a



layer of abstraction that clients need in order to query the database effectively. However, updating your XML schema will be a hassle. Because DB2 doesn't support schema evolution, you will have to physically map your old data to any new schema produced by changing requirements.

After a slow start, due to confusion over what I could and couldn't do without Information Integrator, my tests of DB2 went smoothly. DB2's development interface is very intuitive, stronger than Oracle's and even Sybase's. And despite the absence of XQuery, I found DB2 to have very powerful XML capabilities. It did everything I asked it to do and did it very quickly.

Sybase Pulls a B, Too

I very much like the way Sybase has implemented XML in Sybase ASE. Native XML support is fully integrated with the relational database, and no external programs are necessary to get the functionality.

Sybase ASE has a native XML database that allows storage of shredded, structured, and unstructured XML data. Admins can import the XML data into the database or query XML files directly from the file system. Unfortunately, ASE doesn't support XQuery, so it suffers from the same limitations as

DB2 when it comes to searching XML.

Like Oracle and DB2, ASE allows very rich indexing on XML data, whether shredded or structured. A self-defining indexing mechanism in ASE handles evolving XML formats and structures without requiring any user input. Element values are automatically indexed to speed up predicate searches (the XML equivalent of "select phone from customers where lastname = smith"), and different types of indexes (XML indexes or relational indexes) are chosen by the database depending on the type of query predicate. And queries can be predicated using relational elements, XML elements, or both.

ASE does most of the XML data management for you. Building views, stored procedures, and functions based on XML data is not only easy to code, it's easy to manage as well. However, the lack of schema evolution will hinder growing projects. Lack of schema evolution isn't the only shortcoming in ASE. When creating XML files, ASE cannot perform on-the-fly XSLT translation. The end-user is expected to run the newly created XML files through an XSLT processor afterwards, which hinders any efforts to create XML files in a timely fashion. Any organization that needs to generate XML files on a regular basis should find another solu-

The absence of structured XML support does not leave Microsoft SQL Server dead in the water.

tion. (I find Microsoft .Net to be very good at this task.)

If it weren't for ASE's lack of XSLT and XQuery support, its XML implementation would be in the same class as IBM or even Oracle. Development was extremely easy, and the tools were very intuitive. Sybase has put together an excellent product that should be exploited by any company currently using ASE.

In the speed tests, ASE held its own except when it came to creating XML files, where performance was affected by my having to script XSLT processing. Had ASE included a method for using XSLT, it would have performed well in comparison with the others.

Microsoft Takes a Lesson

Microsoft SQL Server 2000 falls glaringly short of Oracle, IBM, and Sybase when handling XML. As an XML-enabled database, SQL Server only allows you to store XML documents as unstructured objects in the database or to shred the XML data into relational tables.

You might think that the absence of structured XML support would leave SQL Server dead in the water, but that's not entirely so. SQL Server has plenty of functionality that makes it possible to parse and write XML documents. Foremost, you can easily generate XML result sets in SQL Server using the For XML clause in a T-SQL query. This in itself is useful, but is made more so by the ability to scrub the data through an XSLT style sheet along the way.

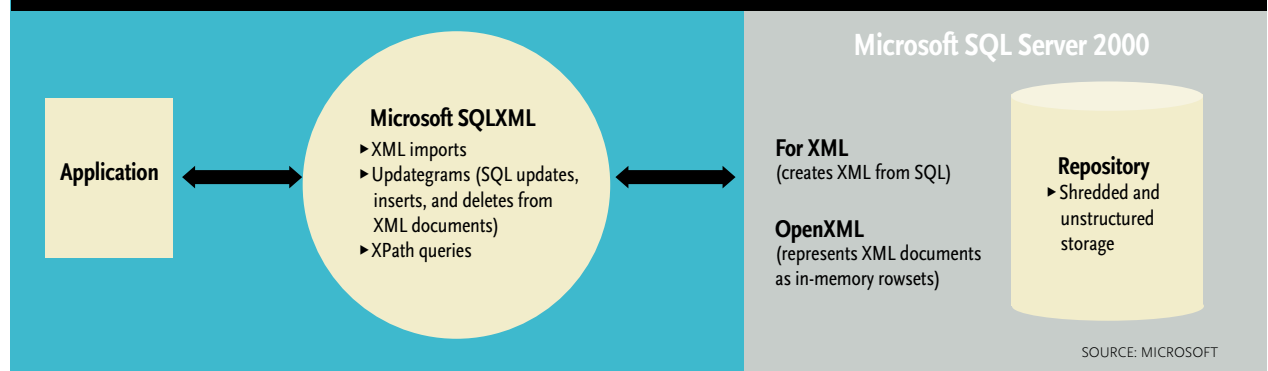
SQL Server has other rich options for creating XML documents, allowing you to slice and dice views and present XML data in almost any form users require. You can also use SQL Server as a sort of bidirectional XML repository by mapping XML documents directly to relational tables through the use of Updategrams, which sit between the XML document and the database and allow you to specify what elements look like before and after a change. This allows you to send updates, inserts, and deletes to SQL Server in the form of an XML document. You receive the results in XML as well.

When reading XML documents, SQL Server gets even more sophisticated. For example, SQL Server can pass XML documents into stored procedures as parameters for shredding into the database. And SQL Server's OpenXML function allows you to generate rowsets from XML documents and insert them into the relational database. In fact, using OpenXML, you have access to some nicely featured methods for manipulating XML data. You can retrieve attributes and elements using flags, and you can define rowset schemas by using column patterns or table names.

However, because SQL Server treats the entire XML document as a single piece of data, the database is not aware that XML elements exist until you start parsing through them in the stored procedure. This isn't the end of the world. You can still build queries and views that leverage the hierarchical relationships of the data in XML files; it just takes a lot of extra work (read: coding) to do so. The next version of SQL Server, due in 2005,

How Microsoft Gets It Done

Most of SQL Server 2000's XML functionality is implemented in Microsoft's SQLXML "middleware," leaving structured XML out in the cold.



will support structured XML storage (see “Will Yukon Strike XML Gold?” below).

How well SQL Server will meet the demands of an XML project depends on what you need to do. If you merely want to import some XML documents into your relational database, and you’re willing to lose the structure of your XML data, then SQL Server will suit you just fine. However, in this scenario, .Net may be a closer match. The .Net tools simply handle some things better, including the creation of XML files.

In my tests, I found that SQL Server speedily imported XML files, but it was the slowest of the group at creating

XML files. It was also slowest at reading large XML files but not by a significant margin.

Relating to XML

All four of these relational databases provide useful XML capabilities. Even SQL Server, which lacks support for structured XML storage, allows you to store, manipulate, and create XML in useful ways.

While Microsoft has a lot of catching up to do, IBM DB2 and Sybase ASE suffer from less significant limitations. Their lack of support for XQuery means less granular XML search capabilities than the others provide. Sybase ASE’s inability to perform XSLT

impromptu transformations means extra time and administrator effort will be necessary to generate XML documents or evolve schema.

Although IBM, Sybase, and even Microsoft offer significant XML functionality, they trail Oracle by a wide margin. Oracle not only does the best job of hiding the complexity of managing XML data, it also offers the richest set of query capabilities and tops it off with extras such as support for schema evolution and WebDAV repositories. If you’re using DB2, Sybase ASE, or SQL Server 2000, you’ll be able to make good use of XML data. ☛

Jennifer Nelson, database administrator at Info Integration, contributed to this article.

Will Yukon Strike XML Gold?

EXPECTED TO SHIP BY MID-2005, YUKON REPRESENTS A MAJOR step forward for Microsoft in the XML arena. Striving to match the native XML storage capabilities in IBM’s, Oracle’s, and Sybase’s relational databases, Yukon will finally store XML data in a structured fashion, along with supporting the shredded and unstructured storage methods already available in Microsoft SQL Server 2000.

Yukon’s approach to preserving the hierarchical data relationships in XML documents differs from that of Oracle, IBM, and Sybase. Instead of using nested tables for storing structured XML, Yukon features what I refer to as a managed BLOB (Binary Large Object) data type. This means that, although the data is being mapped to a regular BLOB data type, the XML data type assigned to the column keeps track of the hierarchy of the XML data. One consequence of the BLOB approach is a 2GB limit on the size of XML files you can store in Yukon.

Will this matter? There are some who think the 2GB limit will be a showstopper for Microsoft in large XML implementations, but I’ll tell you something ... I’ve never seen an XML document even half that size. XML is about finding records in a file, not dumping an entire database to a file and trying to run an XQuery search against it.

Native storage isn’t the only way Yukon improves XML data management. Yukon also offers a simple version of schema evolution. Taking a different route from Oracle, it will allow you to update existing documents by changing the namespace instead of attaching the new schema and scrubbing the existing data through XSLT (XSL Transformation). Microsoft’s method ignores validation for existing data, while Oracle’s method allows you to fill in missing data based on the new schema. In that respect, at least, Oracle’s implementation is more flexible.

You also can validate schema in a number of ways. You can explicitly create an XML schema collection, attach a schema during table creation, or use the cast function to convert data types inside a query. So far, from what I can tell using the Yukon beta, there’s no way to turn off schema validation once you’ve turned it on, and while it seems like a good idea, complete schema validation is expensive. In the interest of conserving time and computing resources, you may not always want to use it, especially against trusted data sources. I’ll certainly be interested in how this shakes out in the final release.

In short, Microsoft seems to be on the right track, but it’s much too early to see how Yukon will pan out. Any Microsoft shop trafficking in XML should be hoping there’s gold in those hills.

— S.M.

NEW SUBSCRIBERS ONLY!

Apply online at: <http://subscribe.infoworld.com>

Priority Code: **MT4PDF**

Form: 18

I WISH TO RECEIVE A FREE SUBSCRIPTION TO **InfoWorld**

☐ 1. Yes ☐ 0. No

SIGNATURE

DATE

A MAILING ADDRESS

Publisher reserves the right to limit the number of complimentary subscriptions. Free subscriptions available in the U.S. (including APO and FPO) and Canada.

NAME

TITLE

COMPANY NAME

DIVISION / DEPT / MAIL STOP

MAILING ADDRESS

CITY / STATE / ZIP / POSTAL CODE

Is the above address a home address?

☐ 1. Yes ☐ 0. No

BUSINESS PHONE (INCLUDING AREA CODE)

FAX NO. (INCLUDING AREA CODE)

E-MAIL ADDRESS

B SPECIAL REQUEST:

You may receive a renewal reminder via e-mail. May we send other information about **InfoWorld** products or services via e-mail? ☐ 1. Yes ☐ 0. No

Reader feedback is important to us. May a member of our editorial team contact you? ☐ 1. Yes ☐ 0. No

1 WHAT IS YOUR ORGANIZATION'S PRIMARY BUSINESS ACTIVITY AT THIS LOCATION? (PLEASE CHECK ONE ONLY):

General Business Industries

- ☐ 01. Defense Contractor / Aerospace
- ☐ 02. Retail
- ☐ 03. Wholesale / Distribution (non-computer)
- ☐ 04. Pharmaceutical / Medical / Dental / Healthcare
- ☐ 05. Financial Services / Banking
- ☐ 06. Insurance / Real Estate / Legal
- ☐ 07. Transportation / Utilities
- ☐ 08. Media (print / electronic)
- ☐ 09. Communication Carriers (telecomm, data comm., TV / cable)
- ☐ 10. Construction / Architecture / Engineering
- ☐ 11. Manufacturing & Process Industries (other than computer-related)
- ☐ 12. Research / Development

Technology Providers

- ☐ 13. Managed Service Provider / Business Service Provider
- ☐ 14. Technology Service Provider (ISP / ASP / MSP, etc.)
- ☐ 15. Computer / Network Consultant
- ☐ 16. Systems or Network Integrator
- ☐ 17. VAR / VAD
- ☐ 18. Technology Manufacturer (hardware, software, peripherals, etc.)
- ☐ 19. Technology - Related Retailer / Wholesaler / Distributor

Government / Education

- ☐ 20. Government: federal (including military)
- ☐ 21. Government: state or local
- ☐ 22. Education

☐ 98. Other _____
(Please specify)

2 WHAT IS YOUR PRIMARY JOB TITLE? (PLEASE CHECK ONLY ONE):

IT / Technology Professionals

- ☐ 01. Chief Technology Officer (CTO)
- ☐ 02. Chief Information Officer (CIO)
- ☐ 03. Chief Security Officer (CSO)
- ☐ 04. Vice President (including SVP, EVP, etc.)
- ☐ 05. Director
- ☐ 06. Manager / Supervisor
- ☐ 07. Engineer
- ☐ 08. Systems Analyst / Programmer / Architect
- ☐ 09. Consultant / Integrator
- ☐ 10. Developer
- ☐ 11. IT Staff
- ☐ 12. Other IT Professional _____
(Please specify)

Corporate / Business Management

- ☐ 13. CEO, COO, President, Owner
- ☐ 14. CFO, Controller, Treasurer
- ☐ 15. Vice President (including SVP, EVP, etc.)
- ☐ 16. Director
- ☐ 17. Manager / Supervisor
- ☐ 18. Other Business Management Title _____
(Please specify)

☐ 98. Other Title _____
(Please specify)

3 PLEASE INDICATE YOUR JOB FUNCTION(S)? (PLEASE CHECK ALL THAT APPLY):

IT / Technology Functions

- ☐ 01. Executive
- ☐ 02. Department Management - IT
- ☐ 03. Research and Development Management
- ☐ 04. Systems / Network Management
- ☐ 05. Management of Enterprise Applications (CRM, ERP, SCM, etc.)
- ☐ 06. Applications Development
- ☐ 07. Consultant / Integrator
- ☐ 08. Other IT Department Management _____
(Please describe)
- ☐ 09. Other IT - Staff _____
(Please describe)

Corporate / Business Functions

- ☐ 10. Executive
- ☐ 11. Department Management - Business
- ☐ 12. Financial / Accounting Management
- ☐ 13. Research and Development Management
- ☐ 14. Sales / Marketing Management
- ☐ 15. Other Department Management
- ☐ 16. Other Department Staff _____
(Please describe)
- ☐ 98. Other _____
(Please describe)

4 HOW MANY PEOPLE ARE EMPLOYED AT THIS ORGANIZATION, INCLUDING ALL OF ITS BRANCHES, DIVISIONS AND SUBSIDIARIES? (PLEASE CHECK ONE ONLY):

- ☐ 01. 20,000 or more
- ☐ 02. 10,000 - 19,999
- ☐ 03. 5,000 - 9,999
- ☐ 04. 1,000 - 4,999
- ☐ 05. 500 - 999
- ☐ 06. 100 - 499
- ☐ 07. 50 - 99
- ☐ 08. Less than 49

5 OVER THE COURSE OF ONE YEAR, DO YOU BUY, SPECIFY, RECOMMEND, OR APPROVE THE PURCHASE OF THE FOLLOWING PRODUCTS OR SERVICES WORTH:

* CONSULTANTS: PLEASE INCLUDE WHAT YOU RECOMMEND FOR YOUR CLIENTS AS WELL AS WHAT YOU BUY FOR YOUR OWN BUSINESS, IF APPLICABLE. IF YOU CANNOT DISTINGUISH BETWEEN THIS AND OTHER LOCATIONS, PUT RESPONSE IN THE FIRST COLUMN.

- | | | |
|----------------------------------|--------------------------------|----------------------------|
| 01. \$100 million or more | 06. \$5,000,000 to \$9,999,999 | 11. \$100,000 to \$399,999 |
| 02. \$50,000,000 to \$99,999,999 | 07. \$2,500,000 to \$4,999,999 | 12. \$50,000 to \$99,999 |
| 03. \$30,000,000 to \$49,999,999 | 08. \$1,000,000 to \$2,499,999 | 13. Less than \$49,999 |
| 04. \$20,000,000 to \$29,999,999 | 09. \$600,000 to \$999,999 | 14. None |
| 05. \$10,000,000 to \$19,999,999 | 10. \$400,000 to \$599,999 | |

Product category	For this location: (write code in box)	For other locations: (write code in box)
Large systems	<input type="text"/>	<input type="text"/>
Client computers	<input type="text"/>	<input type="text"/>
Networking / Telecom (including servers)	<input type="text"/>	<input type="text"/>
Internet / Intranet / Extranet	<input type="text"/>	<input type="text"/>
Security	<input type="text"/>	<input type="text"/>
Storage	<input type="text"/>	<input type="text"/>
Peripheral equipment	<input type="text"/>	<input type="text"/>
Software	<input type="text"/>	<input type="text"/>
Service / Support	<input type="text"/>	<input type="text"/>

Please answer the questions on the following page. 

6**PLEASE TELL US YOUR INVOLVEMENT WITH YOUR COMPANY'S STRATEGIC TECHNOLOGY INITIATIVES (PLEASE CHECK ALL THAT APPLY):**

- ☐ 01. Integrate Technology with company goals
- ☐ 02. Define Architecture
- ☐ 03. Choose Technology Platforms
- ☐ 04. Develop Technology Integration Strategy
- ☐ 05. Test, pilot, implement emerging technologies
- ☐ 06. Scalability Planning
- ☐ 07. Build, Run Web Services
- ☐ 08. Internet / Network Infrastructure
- ☐ 09. Customer Relationship Management
- ☐ 10. External Partnership Management
- ☐ 11. Budgeting
- ☐ 12. Recruitment & Retention
- ☐ 13. Other _____
(Please describe)
- ☐ 99. None of the above

9**ARE YOU INVOLVED IN BUYING, SPECIFYING, RECOMMENDING OR APPROVING THE FOLLOWING TECHNOLOGY SERVICES? (PLEASE CHECK ALL THAT APPLY):**

- ☐ **01. Technology Services**
- ☐ 02. Systems / Application Integration
- ☐ 03. E-Business / Internet / Intranet / Extranet
- ☐ 04. Application Development
- ☐ 05. Application Hosting (ASP)
- ☐ 06. Web Hosting
- ☐ 07. Web Development
- ☐ 08. Security
- ☐ 09. Storage
- ☐ 10. Content Delivery Networks
- ☐ 11. Disaster Recovery / Business Continuity
- ☐ 12. Outsourcing
- ☐ 13. Utility Computing Services
- ☐ 14. Telecommunications
- ☐ 15. Call Center / IT Services
- ☐ 16. Consulting
- ☐ 17. Other Technology Services

7**ARE YOU INVOLVED IN BUYING, SPECIFYING, RECOMMENDING OR APPROVING THE FOLLOWING SOFTWARE? (PLEASE CHECK ALL THAT APPLY):**

- ☐ **01. Enterprise / E-Business Applications**
- ☐ 02. Customer Relationship Management (CRM / eCRM)
- ☐ 03. Enterprise Resource Planning (ERP)
- ☐ 04. Supply Chain / Procurement
- ☐ 05. Business Process Management
- ☐ 06. Business Intelligence / Data Mining
- ☐ 07. Knowledge Management
- ☐ 08. Portals
- ☐ 09. Collaborative Applications / Groupware
- ☐ 10. Project Management
- ☐ 11. Financial / Payroll / Billing
- ☐ 12. E-business / E-commerce
- ☐ 13. Database Management Systems (DBMS)
- ☐ 14. Data Warehouse
- ☐ 15. Manufacturing
- ☐ 16. Asset Management / Software Distribution
- ☐ 17. Performance / Application Management
- ☐ 18. Streaming Media
- ☐ 19. Other Enterprise / E-Business Applications
- ☐ **20. Integration Software**
- ☐ 21. Web Services
- ☐ 22. Web Services Orchestration
- ☐ 23. Application Servers
- ☐ 24. Enterprise Application Integration (EAI) / Middleware
- ☐ 25. Business Process Management
- ☐ 26. Legacy Application Integration Tools
- ☐ 27. Other Integration Software
- ☐ **28. Application Development**
- ☐ 29. Application Development Tools
- ☐ 30. Application Servers
- ☐ 31. Web services
- ☐ 32. Java / J2EE
- ☐ 33. XML
- ☐ 34. .NET
- ☐ 35. Testing Tools
- ☐ 36. Other Application Development Software

10**ARE YOU INVOLVED IN BUYING, SPECIFYING, RECOMMENDING OR APPROVING THE FOLLOWING PRODUCTS OR TECHNOLOGIES? (PLEASE CHECK ALL THAT APPLY):**

- ☐ **01. Networking**
- ☐ 02. LANs (Local Area Networks)
- ☐ 03. WANs (Wide Area Networks)
- ☐ 04. Switches / Routers / Hubs
- ☐ 05. Caching / Load Balancing
- ☐ 06. Grid / Utility Computing
- ☐ 07. E-mail
- ☐ 08. Instant Messaging / Peer-to-Peer
- ☐ 09. Content Delivery Networks
- ☐ 10. Network and Systems Management
- ☐ 11. Traffic Monitoring and Analysis
- ☐ 12. QoS (Quality of Service)
- ☐ 13. VoIP (Voice over IP)
- ☐ 14. Telecommunications
- ☐ 15. IP Telephony
- ☐ 16. Wireless
- ☐ 17. Remote Access
- ☐ 18. Web / Video Conferencing
- ☐ 19. Other Networking
- ☐ **20. Storage**
- ☐ 21. High-end / Enterprise Class Storage
- ☐ 22. Network Attached Storage (NAS)
- ☐ 23. Storage Area Networks (SANs)
- ☐ 24. Storage Management Software
- ☐ 25. IP Storage
- ☐ 26. Direct Attached Storage (DAS)
- ☐ 27. Storage Blades
- ☐ 28. Storage Backup (Tape, Disk, Optical, RAID)
- ☐ 29. Removable / Portable Storage
- ☐ 30. Disaster Recovery
- ☐ 31. Other Storage
- ☐ **32. Security**
- ☐ 33. Anti-Virus / Content Filtering
- ☐ 34. Firewall
- ☐ 35. VPN (Virtual Private Network)
- ☐ 36. Identity Management / Authentication
- ☐ 37. Intrusion Detection
- ☐ 38. Encryption
- ☐ 39. Other Security
- ☐ **40. Internet / Intranet / Extranet**
- ☐ 41. Web Servers
- ☐ 42. Web Development / Authoring Tools
- ☐ 43. Web Performance Management / Monitoring Software
- ☐ 44. Content Management / Document Management
- ☐ 45. Content Delivery Networks
- ☐ 46. Internet Software
- ☐ 47. Other Internet / Intranet / Extranet

8**ARE YOU INVOLVED IN BUYING, SPECIFYING, RECOMMENDING OR APPROVING THE FOLLOWING HARDWARE? (PLEASE CHECK ALL THAT APPLY):**

- ☐ **01. Hardware**
- ☐ 02. Mainframes
- ☐ 03. NT / Windows 2000 / .NET Servers
- ☐ 04. Unix Servers
- ☐ 05. Linux Servers
- ☐ 06. Blade Servers
- ☐ 07. PCs / Workstations
- ☐ 08. Notebooks / Laptops
- ☐ 09. PDAs / Handhelds / Pocket PC / Wireless Devices
- ☐ 10. Other Hardware
- ☐ **11. Peripherals**
- ☐ 12. Laser Printers
- ☐ 13. Inkjet Printers
- ☐ 14. Monitors
- ☐ 15. Flat Panel Displays
- ☐ 16. UPS (Uninterruptible Power Supply)
- ☐ 17. Network Copiers
- ☐ 18. Other Peripherals

11**WHICH OF THE FOLLOWING OPERATING SYSTEMS ARE IN USE OR PLANNED FOR USE AT THIS LOCATION? (PLEASE CHECK ALL THAT APPLY):**

- ☐ 01. Windows XP
- ☐ 02. Windows 2000
- ☐ 03. Windows NT
- ☐ 04. Windows 95/98
- ☐ 05. Windows CE
- ☐ 06. Mac OS (Macintosh)
- ☐ 07. Solaris
- ☐ 08. UNIX
- ☐ 09. Linux
- ☐ 10. MVS, VMS, ESA
- ☐ 11. VM
- ☐ 12. OS 400
- ☐ 13. Netware
- ☐ 14. Palm OS
- ☐ 15. Other OS

**Return this form to InfoWorld,
P.O. Box 3511,
Northbrook, IL 60065-3511 or
FAX to (847) 291-4816.**

You can also apply ONLINE at <http://subscribe.infoworld.com>

