Introduction to XML

Oracle developers and DBAs must master a raft of strategic new standards and technologies. Industry Standard can help you navigate the technology waters. / *By Kelli Wiseth*

For a moment, forget about XML, SGML, HTML, or any other abbreviations, initials, and acronyms. Think instead about the mundane yet frequent annoyance you suffer when submitting a loan application to your bank. You wonder: "My accountant—along with every financial institution I've ever dealt with over the years, including the behemoth that acquired this regional pip-squeak—has all this information somewhere. Why do I have to go through all this again?" Wouldn't it be great if the banks (or the insurance company, or the... fill-in-the-blank) could exchange all this data directly?

Now think about all the information stored in various systems and in various formats around the world. Consider further that every organization, no matter how large or small, is connected to the great global internet, which is ready and waiting to transport this vast amount of information. So why are you sitting with a loan officer filling out yet another mortgage application?

That's one view of the $64,000 question that the Extensible Markup Language (XML) is poised to answer. And not just for data that is already stored in huge databases, but also for new information—e-mail messages, Web pages, invoices, purchase orders, bills of lading, letters of credit between governments—that is being sent every second, from one computer-based system to another. The list is endless, and it covers every industry, from automotive to health-care to human resources, insurance, and travel, and it includes every conceivable format and data type.

John Hersey

WHAT IS XML?

XML adheres to the Standard Generalized Markup Language (SGML) specification, which has been an ISO standard since 1986. SGML has been the basis for many high-end document-processing systems. SGML also provides the means for creating new markup languages for specific purposes—for example, the HyperText Markup Language (HTML) is an SGML-based markup language.

However, although HTML is designed only for displaying hypertext pages in browsers, XML is intended to be used for creating new markup languages for new document types. That means, unlike HTML with its finite set of element types—<TITLE>, <H1>, <H2>, and so on—XML enables users to define new elements for processing information in their particular domain or field. The XML 1.0 specification (available at www.w3c.org/xml) dictates how an XML processor should behave, how to create document types and tag sets, and what it takes to build XML documents.

To learn more, visit the World Wide Web Consortium (www.w3c.org), XML.org (xml.org), and Oracle Technology Network (http://technet.oracle.com).

A bank, for example, could use XML to create a document type called LoanApplication. LoanApplication would need a document-type definition (DTD) containing the entities, elements, attribute lists, and other tags that can be applied to the XML documents that are loan applications: Just as a programmer must declare variables before using them in a program, the creator of an XML document must declare the document type at the beginning of the XML content and must declare the entities, elements, and attributes and establish the relationships among them in a DTD.

The following code is a greatly oversimplified example of the beginning of an XML-encoded loan application. The first line is the XML declaration, the second line is the document-type declaration, and the third through sixth lines are markup declarations for the document type:

```
<?XML version="1.0" encoding="UTF-8"?>
<!DOCTYPE LoanApplication[
<!ENTITY LoanCust(
<!ELEMENT ApplicantName (
<!ELEMENT ApplicantID (
<!ELEMENT ApplicationDate (
<!ELEMENT BankID (
]>
....
<ApplicantID>999-99-9999</ApplicantID>
<BankID>90-78/1211</BankID>
<ApplicationDate>15-JAN-2000</ApplicationDate >
....
```

In this limited example, the definitions are contained at the beginning of the XML-tagged content itself, but a more robust approach is to refer to an external DTD files in a central location and point to them by referencing the DTD within the XML, as in the following declaration:

```
<?XML version="1.0" encoding="UTF-8"?>
<!DOCTYPE LoanApplication "Loans.dtd">
...
```

Although the word *document* pervades discussions of XML, keep in mind that the entities created in the DTDs comprise a hierarchical structure, and when parsed, these entities are objects. In other words, XML isn't limited in scope to being used only with text-oriented documents but is also well suited for object-relational-database information and relational-database information, such as that stored in any Oracle database.

As an example, the following XML code might describe the content from the bank's customer database:

```
<entity id="custType">
<name> cust </name>
<attribute primary_key=1>
<name>cust_no </name>
<type> number </type>
<length> 10 </length>
</attribute>
<attribute>
<name> cust_name </name>
<type> char </type>
<length> 25 </length>
<comment>Customer last name </comment>
</attribute>
</entity>
```

Once a DTD exists, a computer system can use a parser—software that reads through the input stream, checks for conformity to the document type defined, and extracts the data (the content) from the tags. When the XML processor parses an input stream, it maps the result to the document-object model (DOM)—a tree-based hierarchical object—for further manipulation by the application. Alternatively, you can use the Simple API for XML (SAX) interface, which uses an event-driven programming model. Each has appropriate uses in particular environments.

After the system has parsed the XML input into the DOM object nodes (hierarchical, parent/child node relationships), an application (or multiple applications) can access the objects for further processing. Such processing might include rendering the content for a variety of targets, such as a Web browser, e-mail, pager display, cell phone, or formatting for input and integration with existing enterprise resource planning (ERP) systems or even legacy applications. The ability to parse once and render to many different media and application targets depends in large part on applying style sheets created by using the Extensible Stylesheet Language (XSL) specification, another emerging standard in the XML suite.

WHAT ABOUT XSL, XSLT, XCETERA?

First adopted by the World Wide Web Consortium (W3C; www.w3c.org) in February 1998, XML has evolved from a single markup language to encompass the concept of a complete framework that includes nearly 20 proposals, recommendations, and specifications to complement the core XML language specification. (Some of these are discussed briefly in this article, but refer to the W3C for complete information.) The W3C guides the development of XML and related recommendations, proposals, and specifications; provides a library of technical material and working drafts; and sponsors the working groups that continue in each of several emerging areas.

◆ The XSL specification defines an XML vocabulary for specifying formatting semantics.

◆ An adjunct to XSL, the XSL Transformation (XSLT) language, transforms XML documents into other types of XML documents. (An XSLT processor is built in to the Oracle XML processor.) XSLT is intended for use in conjunction with XSL style sheets for rendering an XML stream into the format for the appropriate target.

◆ Currently in working-group status are proposals for the XML Schema definition language, which will provide a more global and robust way of dealing with DTDs. In essence, in order for there to be transparent interoperability, there must be standard agreement about the structure and format of files.

- The XML Namespace specification, a formal recommendation of the W3C since early 1999, allows multiple applications to share a markup vocabulary. An XML document is associated with a namespace by use of a uniform resource identifier, typically a URL.

THE ORACLE INTERNET PLATFORM AND XML

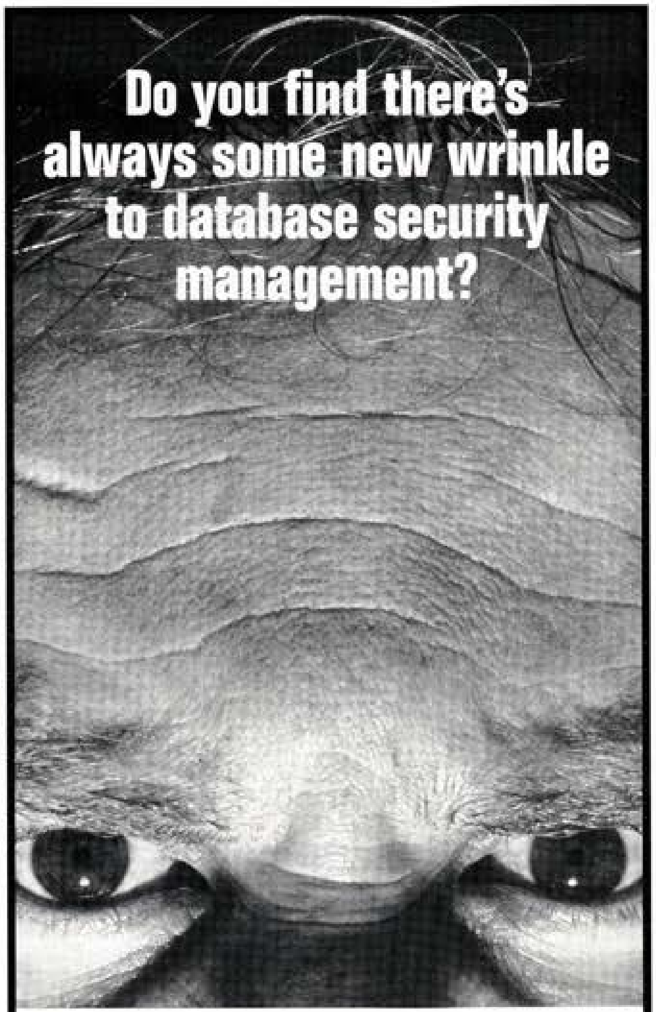
Oracle has been a leader in the various XML industry initiatives, as an active member of the W3C and a founding sponsor of XML.org, a self-supporting community resource that serves as a reference site for XML vocabularies, DTDs, schemas, and namespaces. The company is also actively working on the W3C's XML Schema standardization effort recognizing that schema standards are necessary for achieving long-term global business-to-business integration and application integration.

Beyond Oracle's work on XML standards committees, the Oracle Technology Network (OTN; http://technet.oracle.com) provides developers with all the software components (free of charge) they need to begin creating XML applications. Specifically, these software components include XML parsers for Java (versions 1.1 and 2), C, C++, and PL/SQL. Oracle's XML parsers are compliant with the W3C standard for XML 1.0 and support the SAX interface in addition to the XML 1.0's required DOM programming model.

You can also download Oracle's XML SQL Utility for Java from OTN. This utility greatly simplifies the task of transforming database records into XML elements and vice versa. The SQL utility can generate an XML document from SQL queries and can write data from XML input into a database table or view. Also available are the Oracle Class Generator for Java, which creates classes from DTDs to enable the programmatic construction of XML documents, and sample Java servlets that highlight some of the functionality of Oracle's XML technology and capabilities.

You can use these facilities today to begin working with XML, but Oracle is integrating all of these components natively into the Oracle internet platform. And start thinking of ways that you as a developer can use XML to enable seamless transparent application integration.

Maybe the next time you need to fill out a loan application, your bank will simply dip in to its existing data stores—or poll other banks using an XML-formatted query—and populate a Web page for you to simply confirm or change the data to submit the new loan application.


Do you find there's always some new wrinkle to database security management?

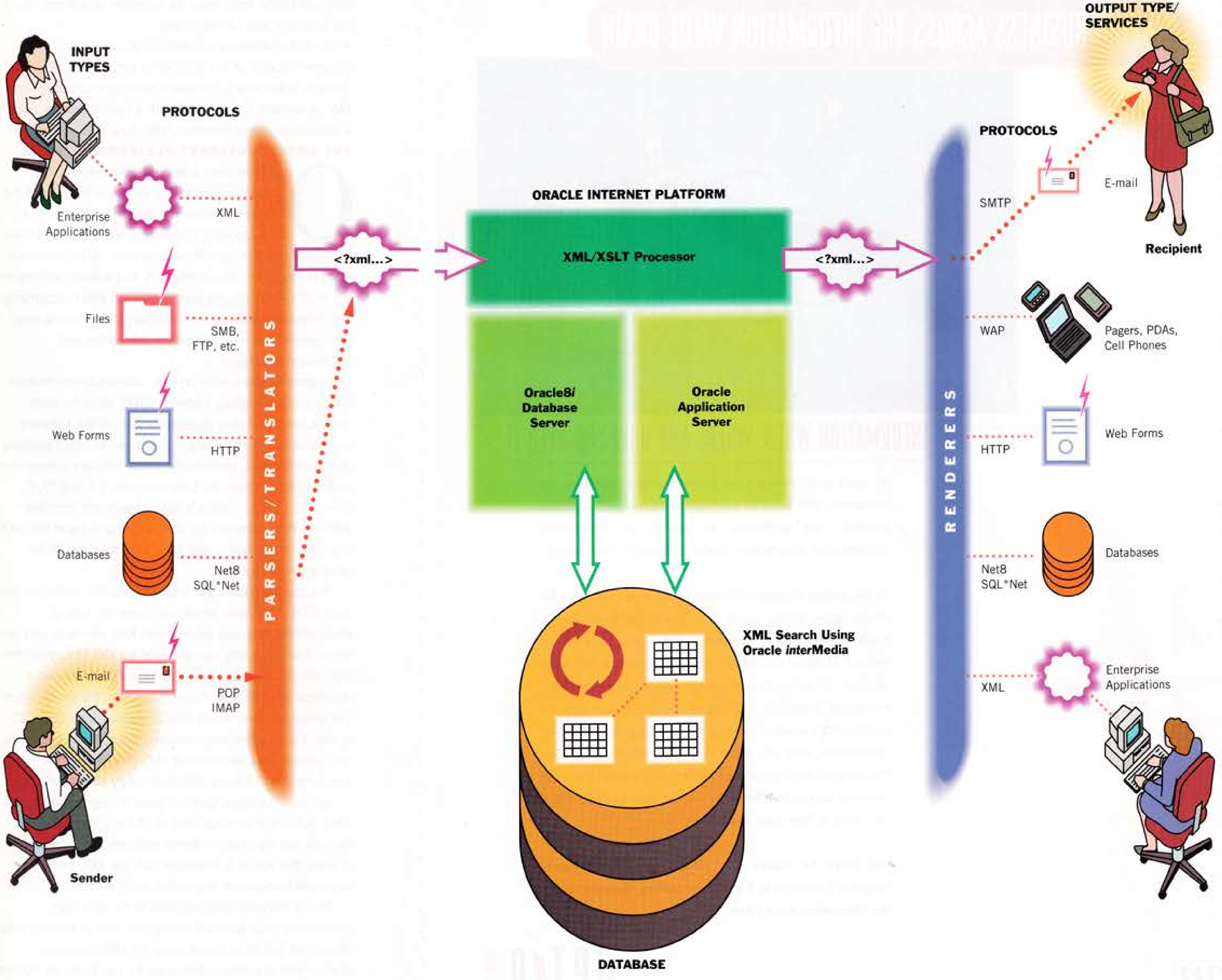
INTERNET ARCHITECTURE

XML in the Oracle Internet Platform

The problem of exchanging data among disparate systems without losing the meaning of the information has long been a thorn in the side of IT. The problem has been the inability to separate data content from data use. Now, with XML—Extensible Markup Language—embedded directly in the Oracle internet platform, a solution is on the way. XML enables complete separation of storage and management of data from production and consumption.

Oracle is implementing XML across the entire technology stack. For developers, that means you'll be able to build new, internet-enabled systems that can effectively deal with any input and target any device, over any network. You can also leverage existing data stores to build dynamic, open applications that will work on virtually any device or hardware platform. Components from Oracle running in either the Oracle8*i* database or Oracle Application Server can take any input over any protocol, parse or convert the input to XML, and then output XML through a variety of renderers to virtually any device. For example, you could develop an application that accepts traditional e-mail messages and then converts and renders them for delivery via a simple mail transfer protocol (SMTP) gateway; for integration with enterprise applications; or for transmission using wireless area protocol (WAP) for output on a pager's display.

For additional information and to download all the tools you'll need to get started with XML in your own applications, visit Oracle Technology Network. ◆



Jeffery West