

XML Support in Oracle 9iAS

*An Oracle White Paper
March 2001*

XML Support in Oracle 9iAS

Executive Overview	3
Introduction	3
The New Application Architecture Requirements	4
Business Partner Requirements.....	5
B2C and B2B Requirements.....	5
Content Provider Requirements.....	5
Service Provider Requirements	5
The Oracle XML-enabled Internet Platform	6
XML Developer's Kits	6
XML Parsers.....	6
XSL Processors	7
XML Class Generators.....	7
XML Transviewer JavaBeans	7
XML SQL Utility.....	9
XSQL Servlet.....	10
XML-Enabled Application/Web Servers	11
XML-enabled Deployments.....	11
XML-Enabled Tools.....	12
Conclusion: XML's Impact on Application Development	12

EXECUTIVE OVERVIEW

As the speed of business increases, the requirement for data interchange without human intervention becomes a high priority requirement. Integrating XML with Enterprise Databases and Application Servers and XML-enabling their applications gives businesses the infrastructure to satisfy this increasing demand for access to and exchange of information.

Oracle9i Application Server along with the Oracle8i database provides a platform for natively handling XML data and documents to Internet standards and provides the development resources and interfaces to fully integrate this support into business applications.

INTRODUCTION

The modern business can no longer depend upon paper to convey transactions. The Internet has brought competition into local businesses from across the nation and even the world. Today's business must be able to not only sell competitively but also purchase their inventory within a wholesale competitive arena.

While databases have been a mainstay in business for the last decade data entry has been mostly done by keyboard entry. This is because it was the rare cross-business system that was sufficiently homogenous to conduct transactions electronically. Extensible Markup Language, XML, in conjunction with the Internet offers the first real paradigm shift in the transaction of business across heterogeneous systems commonly in place today.

Real e-business involves more than just putting up an online storefront. Orders coming in through a web site should flow directly to back-office order entry, inventory, and fulfillment systems. Suppliers and other vendors should be integrated into the production process to improve efficiency and reduce costs. And inside the firewall, self-service web applications for operational functions such as human resources and expense processing must be integrated with existing databases and applications.

To take advantage of e-business and the opportunities it presents, applications will need to connect employees with each other, with customers, and with suppliers and other partners. The ideal application infrastructure must support not only traditional point-to-point enterprise application integration (EAI), but also the newer, "loosely-coupled" services that are the future of e-business. Examples of such services include online bidding to supply production materials, syndicated information feeds flowing to enterprise information portals, catalog publishing and order processing via vertical industry portal sites that aggregate vendors, and many others.

Oracle's entire product line is based on Internet computing for e-business. Together, Oracle's database and application server, middle-ware, development tools, content-management features, and system administration and integration capabilities provide an enterprise-class infrastructure for Internet applications—the Oracle9i Internet Platform. Key products for developers include Oracle8i, the Internet database; Oracle9i Internet Application Server, an open, standards-based server for deploying three-tier applications; and Oracle9i Internet Developer Suite, a powerful programming environment for building and deploying Internet applications.

THE NEW APPLICATION ARCHITECTURE REQUIREMENTS

As companies increasingly depend on the Internet to conduct business, it is no longer satisfactory to run applications that depend upon paper to exchange data. These applications must be able to exchange data without human intervention and at the speed of the Internet. Figure 1 shows how different business can make use of XML to integrate their application stacks.

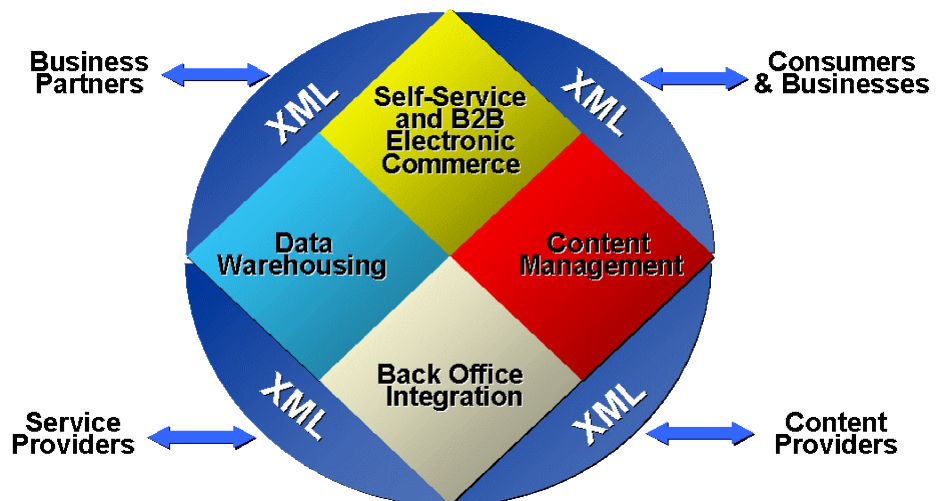


Figure 1: XML-Enabled Real Use Solutions

Each of these business areas has a set of requirements in these application areas which can be met through the family of XML technologies.

Business Partner Requirements

Business partners need to be able to exchange data without the errors and delays that come from “re-keying” invoices, purchase orders, etc. They need this data to be exchanged electronically without imposing strict formats or business logic on their trading partners. They also need to seamlessly integrate real-time data with historical data in data warehouses. XML and the related technologies of XSL, DOM, SAX, and XPath when implemented in enterprise-level servers and applications can meet these requirements. In fact, businesses are setting up trading partner exchanges with this technology.

B2C and B2B Requirements

The movement of business transactions online has brought with it a set of requirements across areas of CRM, e-business and content management. Communication must be over the Internet and both the product information and transaction must be delivered on-demand. This requires inventory databases as well as catalogs to be brought online and the customer experience tailored to his access device of choice. Once again, XML’s ability to separate content from presentation along with cross platform data encapsulation can deliver on these requirements.

Content Provider Requirements

More and more companies, including Oracle, are depending less and less on delivering content through the printed page. Not only is there the issue of the currency of the material, but also customers and subscribers are demanding content their way. Data must be able to be dynamically rendered from up-to-date sources across a wide range of presentation devices. Here again, XML coupled with XSL can not only deliver the current content but can be used as a custom control and presentation technology across many platforms and languages.

Service Provider Requirements

The pervasiveness of computers and the Internet in the operation of even small businesses has brought a significant technology requirement to the everyday operation of businesses. This expertise is expensive to acquire and maintain in-house. As a result service providers have sprung up to provide businesses with professionally managed gateways to the Internet as well as the hosting of applications. Therefore, these service providers need easily customizable applications that can be distributed across the Internet. With XML’s ability not only to encapsulate data but controls, configurations, documentation, etc. these service providers can quickly

respond to their customer requirements and minimize the technical expertise required from them.

THE ORACLE XML-ENABLED INTERNET PLATFORM

To deliver on all of these varied these requirements companies need access to an entire XML-enabled Internet Platform. This development and deployment platform must deliver standards-based functionality that can be flexibly deployed in a performant and scalable manner. To integrate into existing systems, this platform must support multiple languages, operating systems, and mainframes to PCs to PDAs to pagers. Oracle has spent the last two years developing just such a platform which includes XML infrastructure components, databases, application servers, and platform services.

XML Developer's Kits

Oracle has packaged its generic XML components and utilities into a series of XML Developer's Kits or XDKs. These XDKs are available in the platform independent languages of Java, JavaBeans, and PL/SQL as well as the platform-dependent languages of C and C++. As XDKs these components are included with a variety of Oracle products including Oracle 8i, Oracle9i Internet Application Server, Oracle Internet Developer's Suite and JDeveloper.

The Java XDKs can be used with JDK 1.1.6 or greater. The C/C++ XDKs include the appropriate dependent CORE and NLS libraries. The PL/SQL XDK currently requires the Java parser but in 9i will be built upon the C one. Included with the XDKs are the following W3C standard components.

XML Parsers

The foundation component of any XML development is the XML parser. The parser provides programmatic access to XML documents enabling them to be used in many ways within an application. Oracle's parsers support the two industry-standard specifications for XML document access – DOM and SAX.

The DOM (Document Object Model) interfaces provide fine-grained access to an in-memory representation of the XML document. This is a tree representation of nodes and applications can perform operations on it such as retrieving, modifying, and deleting nodes or their content. This W3C specification is supported in all of the Oracle parsers at the 1.0 level.

The SAX (Simple API for XML) interfaces are designed for event-driven access to XML documents. Programmer's can register document events, start of element, start of attribute, etc., with a listener enabling documents

of virtually any size to be streamed. These interfaces are also supported by all but the PL/SQL parser at the 1.0 conformance level.

Along with the DOM and SAX support, the parsers include the full internationalization support provided by Java and Oracle's NLS libraries. Production versions of the parsers are included in Oracle 8*i*, 9iAS, and from the Oracle Technology Network. The Java version is also included in JDeveloper 3.1 and greater.

XSL Processors

The eXtensible Stylesheet Language, XSL, is the XML-based technology responsible for transforming or formatting an input XML document into a different output one or rendered into virtually any text-based output. Integrated into all of the Oracle XML Parsers are XSL Processors that perform these transformations. These processors are conformant with the XSLT 1.0 W3C Recommendation and have high performance architecture along with stylesheet caching to improve throughput. The XSL Processors are available in Java, C, C++, and PL/SQL and currently production. They are included Oracle 8*i*, 9iAS and from the Oracle Technology Network. The Java version is also included in JDeveloper 3.1 and greater.

XML Class Generators

The XML Class Generators in C++ and Java are utilities that create C++ and Java Classes from DTDs to enable the programmatic construction of XML documents. The Class Generator works in conjunction with the Oracle XML Parser, and boasts complete character set support and optional validation mode for ease of debugging. Documents created by these classes are fully compliant with the XML 1.0 W3C standard. These classes can be easily used in applications, applets, JSPs and CGI to add XML generation capability. Both the Java and C++ versions are production and included in Oracle 8*i*, 9iAS, and from the Oracle Technology Network.

XML Transviewer JavaBeans

The Oracle XML Transviewer JavaBeans are a set of visual and non-visual beans that are useful in creating a variety of XML-enabled Java applications or applets. The bean encapsulation includes documentation and descriptors that may be accessed directly from Java Integrated Development Environments like JDeveloper. The visual beans extend JPanel and thus are dependent upon Swing. The following is a list of the current beans available on the Oracle Technology Network and Oracle 9*i*. All but the DBView and DBAccess beans have been available in Oracle 8*i*, iAS, and JDeveloper.

DOMBuilder Bean

The DOMBuilder Bean is a non-visual component that encapsulates the DOMParser and provides asynchronous XML document parsing. This enables multiple document parsing or control to be returned immediately after starting a parse to the calling application making GUI applications more user-friendly.

XSL Transformer Bean

The Transformer Bean is a non-visual bean that encapsulates the XSL Processor. It accepts an XML document and applies the transformation specified by an input XSL stylesheet to create and output file taking full advantage of the performance enhancements such as stylesheet caching.

XMLSourceViewer Bean

The XMLSourceViewer Bean is a visual bean that displays XML and XSL-formatted files with color syntax highlighting for easy viewing and editing. It permits selection of foreground and background colors as well as the specific font to be used when displaying XML types including elements, attributes, comments, CData, etc.

XMLTreeViewer Bean

The XMLTreeViewer Bean is a visual bean that displays XML-formatted files graphically as a tree. The tree represents the in-memory DOM of the document. These branches and leaves of this tree can be expanded and collapsed with a mouse.

XMLTransformPanel Bean

The XMLTransformPanel Bean is a multi-function visual bean that encapsulates the above beans in an application component for retrieving, transforming, viewing, and editing XML files. It provides a file interface to XML documents and has a built-in HTML viewer for viewing the output of a stylesheet transformation of an XML file.

PURCHASE ORDER Order No. 3001

TO	ACME Products	
ADDRESS	100 Main St., Anytown	DATE Jan 1, 1999
SHIP TO	Joe's Gym	DEPT NO. A-100
ADDRESS	300 Wall St., Anytown	FOR Jane Smith

PLEASE NOTIFY US IMMEDIATELY IF YOU ARE UNABLE TO SHIP COMPLETE ORDER BY DATE SPECIFIED

	QUANTITY	PLEASE SUPPLY ITEMS LISTED BELOW	PRICE
1	1	ACME Exerciser Pro	\$1.00
2	4	Thigh Master	\$49.95

DATE REQUIRED	Jan. 30, 1999	HOW SHIP	FedEx
TERMS	Net_20	PURCHASING AGENT	John Doe

Figure 2: TransformPanel Bean

XML SQL Utility

Modern databases understand SQL and JDBC. As such the XML Utilities in Java have been created to generate XML documents as text or DOM trees. SQL queries from the XQSL Servlet, Java stored procedures, or even the command line can be sent to the database with the results formatted as XML based upon the internal structure of the schema. These classes accept the parameters and SQL statement between the <query> tags and connect to the database with the connection string passed as an attribute of the query tag. The ResultSet is returned in XML and is based upon the schema where column names become element tags and rows have unique ID attributes. Additionally, an XML-to-SQL store class has been provided to parse and insert data from an XML document to its respective schema based upon columns representing element tag names. This component is available in Java and PL/SQL in Oracle 8i and in Oracle9iAS.

Besides supporting selects, inserts, updates, and deletes, the XML SQL Utility can output the Document Type Definition (DTD) of the schema associated with these operations. This data model can then be input into the XML Class Generator described previously to deliver the Java or C++ classes that can then be used in an application, applet, JSP, CGI, etc. to create conforming XML documents that can be sent to the database and their data decomposed and inserted into the schema.

XSQL Servlet

The XSQL Servlet is a server component written to assist in producing dynamic XML documents from one or more SQL queries of data objects. It does this by processing an xsql file which is simply an XML file with a specific structure. The following is an example of an xsql file that searches for insurance claims submitted by Mr. Doe. Note that the query may be written as regular SQL or, as in this case, using object dot notation.

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="claim.xsl"?>
<query connection="xmldemo">
  select value(c) as Claim from insurance_claim_view c
  where c.claimpolicy.primaryinsured.lastname = 'Doe'
</query>
```

The XSQL Servlet uses Oracle's XML Parser to process this file and pass any XSL processing statements to its internal XSLT Processor while passing the parameters and SQL statements between the <query> tags to the XML SQL Utilities. Results from those queries are then received as either XML formatted text or a JDBC ResultSet object. If necessary, the query results may be further transformed into any desired format using the built-in XSLT processor. The code below shows how the XSQL Servlet utilizes the "media" tag in HTTP request headers to associate a stylesheet to the requesting browsers thereby allowing the website designer to dynamically render content in accordance to the display capabilities of the user.

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" media="MSIE 5"
href="doyouxml-msie.xsl" ?>
<?xml-stylesheet type="text/xsl" media="Mozilla/3.0"
href="doyouxml-hw.xsl" ?>
<?xml-stylesheet type="text/xsl" media="Lynx"
href="doyouxml-lynx.xsl" ?>
<?xml-stylesheet type="text/xsl" href="doyouxml.xsl" ?>
<datapage connection="demo" xmlns:xsql="urn:oracle-
xsql">
```

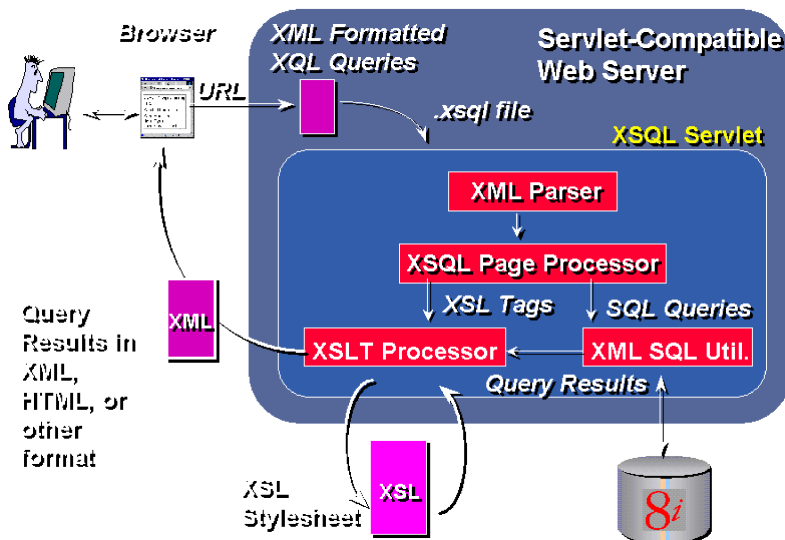


Figure 3. XSQL Servlet Data Flow

Figure 3 shows the data flow through the XSQL Servlet and its use of the other XDK components. The XML Parser is used by the XSQL Servlet to parse the xsql files as well as to receive and render as necessary the output from the XML SQL Utilities. As with the servlet, the Parser can be loaded into the Java VM (virtual machine) in Oracle8i, run from a middle tier server such as Oracle Internet Application Server, or run on the client as a Java bean. Through its DOM and SAX support, it can accept the ResultSet from the XML SQL classes and provide an XML object tree or text stream to the servlet for output.

XML-Enabled Application/Web Servers

Whether as a web server or deploying applications, Oracle9i Application Server is a very capable and scalable server. Now shipping with a complete array of development and management tools, 9iAS, is the perfect choice to mate with the Internet-ready database in Oracle 8i. The XML support across the server and tools integrates seamlessly with the support in 9iAS as many of the components were designed to work equally well on any tier.

XML-enabled Deployments

Just as in Oracle 8i the Oracle XDKs are included in 9iAS to aid in deployment of XML-enabled applications and web sites. This flexible deployment capability permits developers to design their applications placing the data processing and business logic on the most appropriate tier to deliver the required scalability, security, performance and reliability. In fact Oracle Business Components for Java is an XML-enabled encapsulation of just this functionality. The flexible license and the range of supported XML technologies deliver the Internet platform that

companies need to productively build and cost-effectively deploy reliable and scalable Internet applications exploiting XML.

XML-Enabled Tools

Included in 9iAS are a set development and management tools that been XML-enabled. JDeveloper in its 3.2 version handles XML files natively, can readily retrieve XML-formatted data from Oracle 8i and rapidly develop once and deploy anywhere XML applications in 9iAS utilizing the included XDKs.

Also included are Oracle's business intelligence tools. Oracle Reports, starting in 6i can output into XML format which can be readily rendered using the included XSL Processor to a full range of easily customized formats. Discoverer 3i not only accepts XML as an input file type but utilizes it as a component properties descriptor and XSL stylesheets to render custom presentations.

CONCLUSION: XML'S IMPACT ON APPLICATION DEVELOPMENT

Both software vendors and end-user companies around the world have a huge investment in their existing relational and object-relational applications. For the most part, these applications work well. However, if they can represent their data as XML and can accept XML for data input, then their data can be extended and made much more useful and flexible. This approach can be used to "XML-enable" most of the important business information in the world today. In addition, the ability to parse XML data and store it in object-relational tables without loss of structure allows an object-relational database such as Oracle8i to integrate the administration, organization, and search of virtually all business information. The Oracle Internet Platform is well positioned to continue as the foundation for building enterprise applications, while embracing all the benefits of working with XML-formatted data.

Over the past 15 years, businesses have overwhelmingly chosen relational databases as the infrastructure to run their enterprise applications. This is due to the flexibility and broad applicability of the technology. XML is now becoming popular for largely the same reasons. In the preceding sections, the requirements for enterprise data management and enterprise application deployments made possible by XML have been discussed showing that modern object-relational databases, such as Oracle8i, and application servers such as Oracle9iAS are best suited for the task. By using the Oracle XDK components, it is relatively simple to fully use XML data and documents in applications and content management. Moving forward, Oracle8i coupled with similarly enabled Oracle9i Application Server can become the integration point and repository for all enterprise information.



XML Support in Oracle 9iAS
March 2001
Author: Mark V. Scardina

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com

Oracle Corporation provides the software
that powers the internet.

Oracle is a registered trademark of Oracle Corporation. Various
product and service names referenced herein may be trademarks
of Oracle Corporation. All other product and service names
mentioned may be trademarks of their respective owners.

Copyright © 2001 Oracle Corporation
All rights reserved.