SECURITY IS THE WATCHWORD THESE DAYS

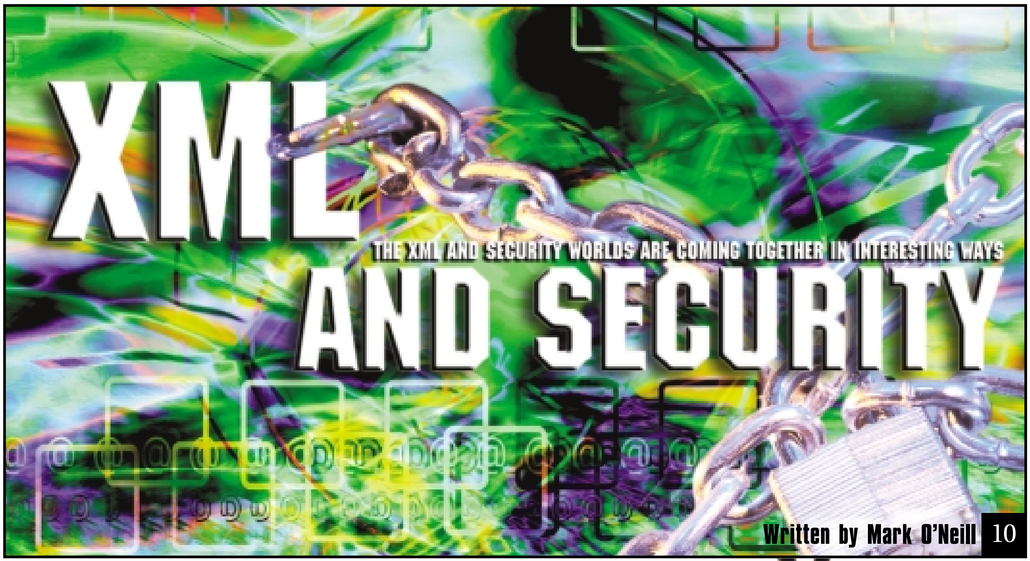
XML JOURNAL

The World's Leading XML Resource

Volume: 2 Issue: 12

XML-JOURNAL.COM

SYS-CON MEDIA



XML AND SECURITY

THE XML AND SECURITY WORLDS ARE COMING TOGETHER IN INTERESTING WAYS

Written by Mark O'Neill 10

XML JOURNAL

DEPARTMENT EDITORS
EDITOR-IN-CHIEF: Ajit Sagar
EDITORIAL DIRECTOR: Jeremy Geelan
E-BUSINESS EDITOR: Israel Hilerio
JAVA TECHNOLOGY EDITOR: David Johnson
PRODUCT REVIEW EDITOR: Jim Milbery
WEB SERVICES EDITOR: Graham Glass
EXECUTIVE EDITOR: M'lou Pinkham
MANAGING EDITOR: Cheryl Van Sise
EDITOR: Nancy Valentine
ASSOCIATE EDITORS: Jamie Matusow
Gail Schultz
Brenda Greene
ASSISTANT EDITOR: Lin Goetz

WRITERS IN THIS ISSUE
Russ Atkind, Mark Cyzyk, Tod Emko, Sean Harvey,
Roy Hoobler, Coco Jaenicke, Siet Lang Lai, Jim Milbery,
Bob McWhirter, Anita Nicholson, Mark O'Neill, Ajit Sagar,
David Silverlight, Simeon Simeonov, Joseph Smith

SUBSCRIPTIONS
SUBSCRIBE@SYS-CON.COM
For subscriptions and requests for bulk orders,
please send your letters to Subscription Department
Subscription Hotline 800 513-7111
Cover Price: $6.99/issue
Domestic: $77.99/yr (12 issues) Canada/Mexico: $99.99/yr
all other countries $129.99/yr
(U.S. Banks or Money Orders)

EDITORIAL OFFICES
SYS-CON MEDIA
135 CHESTNUT RIDGE ROAD, MONTVALE, NJ 07645
TELEPHONE: 201 802-3000 **FAX:** 201 782-9637
XML-JOURNAL (ISSN# 1534-9780)
is published monthly (12 times a year)
by SYS-CON Publications, Inc.
Periodicals postage pending
Montvale, NJ 07645 and additional mailing offices.
POSTMASTER: Send address changes to:
XML-JOURNAL, SYS-CON Publications, Inc.,
135 Chestnut Ridge Road, Montvale, NJ 07645.

WRITTEN BY **AJIT SAGAR** EDITOR-IN-CHIEF

E-Business as Usual?

I just got back from the Web Services Edge/XMLEdge conference hosted by **SYS-CON Media** in Santa Clara, California. Although attendance was lower than usual, this wasn't unexpected and the show was quite well received by both vendors and attendees. Look for the XMLEdge show report later in this issue.

It's quite interesting to see how XML vendors are repositioning themselves as Web services vendors to latch on to the latest buzz in the industry. Although touted as the components of the next platform for conducting e-business, Web services still have a long way to go. And not everything will be "Web service–enabled."

For that matter, not everything is going to be Web-enabled. Several facets of business applications simply aren't suited for the Web. One of the most ludicrous statements I've seen is that "Web services are the killer app for XML." This is inaccurate. Web services are not an application. Rather, they're components with well-defined interfaces that can be used to build applications.

The underlying technology that enables the creation of these components is XML. Since XML is basically a technology used for expressing data using metadata, Web services is the result of leveraging XML to express a service. The standard for expressing services is WSDL (Web Services Definition Language). XML-based message envelopes (in the form of SOAP messages) can add transport and message-passing information to the raw XML data. UDDI is used to find registered services. Now, these services may be combined to create some novel applications that leverage the Web, and some of them may end up as the next breed of killer apps for the Web. However, some of the areas that XML will continue to address outside the realm of Web services are:

- Non-Web service–based applications
- Data formatting and translation
- Data aggregation and normalization
- Content management and personalization
- Business process definitions
- Web service frameworks and toolkits
- Enterprise messaging for EAI and B2Bi
- Application configuration and deployment

This isn't an exhaustive list; it merely mentions some of the areas that XML addresses and will continue to address while Web services establish their role in Web-based applications.

One of the areas emerging in XML data transport is the concept of e-business messaging. Messaging vendors that provide JMS implementations are offering environments that avoid the complexity and cost of full-blown application servers by providing an environment for exchanging message-based data. The transport paradigm usually supports both queue-based point-to-point as well as publish/subscribe–based messaging. In e-business messaging a message format is the basis of data exchange between two parties. XML makes this easier by deciding many data- formatting issues with the addition of other XML standards such as RosettaNet and ebXML. No additional work is required to come up with a message format.

Several initiatives in the standards space target e-business messaging using XML. XML messaging has become an important component of software development platforms such as J2EE and .NET. These platforms provide standards and APIs for XML-based messaging. For example, Sun's JAXM (Java API for XML Messaging) initiative enables applications to send and receive document-oriented XML messages using a pure Java API. JAXM implements SOAP 1.1 so that developers can focus on building, sending, receiving, and decomposing messages for their applications instead of programming low-level XML communications routines.

Software platforms are definitely evolving to create tools and environments for developers to avoid dealing with writing and managing raw XML documents. Many of these environments are geared toward creating Web services. As the industry evolves around XML, many of the "raw XML" environments will move to a lower layer in enterprise solutions and act as enablers for applications using Web services and business process workflows.

AJIT@SYS-CON.COM

AUTHOR BIO

*Ajit Sagar is the founding editor and editor-in-chief of **XML-J** as well as the J2EE editor of **Java Developer's Journal**. A senior solutions architect with VerticalNet, he is an expert in Java, Web, and XML technologies.*

SYS-CON MEDIA
135 Chestnut Ridge Rd., Montvale, NJ 07645
TELEPHONE: 201 802-3000 **FAX:** 201 782-9607

PUBLISHER, PRESIDENT, and CEO
Fuat A. Kircaali fuat@sys-con.com

VP, BUSINESS DEVELOPMENT
Grisha Davida grisha@sys-con.com

ADVERTISING
Senior VP, Sales & Marketing
Carmen Gonzalez carmen@sys-con.com
VP, Sales & Marketing
Miles Silverman miles@sys-con.com
Advertising Sales Director
Robyn Forma robyn@sys-con.com
Advertising Account Manager
Megan Ring megan@sys-con.com
Associate Sales Managers
Carrie Gebert carrie@sys-con.com
Kristin Kuhnle kristin@sys-con.com
Sales Assistant
Alisa Catalano alisa@sys-con.com

EDITORIAL
Executive Editor
M'lou Pinkham mpinkham@sys-con.com
Managing Editor
Cheryl Van Sise cheryl@sys-con.com
Editor
Nancy Valentine nancy@sys-con.com
Associate Editors
Jamie Matusow jamie@sys-con.com
Gail Schultz gail@sys-con.com
Brenda Greene brenda@sys-con.com
Assistant Editor
Lin Goetz lin@sys-con.com

PRODUCTION
Vice President, Production
Jim Morgan jim@sys-con.com
Art Director
Alex Botero alex@sys-con.com
Assistant Art Directors
Cathryn Burak cathyb@sys-con.com
Louis F. Cuffari louis@sys-con.com
Graphic Designers
Richard Silverberg richards@sys-con.com
Aarathi Venkataraman aarathi@sys-con.com

SYS-CON EVENTS
VP, SYS-CON Events
Cathy Walters cathyw@sys-con.com
Conference Manager
Michael Lynch mike@sys-con.com
Sales Executives, Exhibits
Michael Pesick michael@sys-con.com
Richard Anderson richard@sys-con.com
Show Assistant
Niki Panagopoulos niki@sys-con.com
Registration Assistant
Jaclyn Redmond jaclyn@sys-con.com

JDJ STORE.COM
JDJStore Manager
Anthony D. Spitzer tony@sys-con.com

WEB SERVICES
Webmaster
Robert Diamond robert@sys-con.com
Web Designers
Stephen Kilmurray stephen@sys-con.com
Christopher Croce chris@sys-con.com

ACCOUNTING
Chief Financial Officer
Bruce Kanner bruce@sys-con.com
Assistant Controller
Judith Calnan judith@sys-con.com
Accounts Receivable
Jan Braidech jan@sys-con.com
Accounts Payable
Joan LaRose joan@sys-con.com
Accounting Clerk
Patti Del Vecchio patti@sys-con.com

SYS-CON MEDIA

WRITTEN BY **COCO JAENICKE**

Code Reuse: From Objects to Components to Services

Now that the age of limitless optimism is over and it's trendy to be cynical, I hear many Web services cynics remark that there's nothing new here. They're just components. Been there, done that, and in fact we called it CORBA (or COM). This leads to the inevitable questions about what truly is new and different, and what is empty hype for yesterday's news.

In the Beginning...

Once upon a time, programmers analyzed their needs, wrote code, and solved their problems. Life was good. Then scientists discovered the object and promised code reuse and shorter development cycles. C++ was the legend of the day…except for two problems: C++ was hard to use and the compiled objects were proprietary. So scientists went back to work and discovered the component. Code reuse and short development cycles forever! Except for two things: components are hard to use and are proprietary.

After that, scientists probably spent a fair amount of time playing games on the Web. But they did eventually tear themselves away and emerged with a new form of reusable code called the *Web service*. Yes, much of the same verbiage we heard in the past, but there are two things that Web services is getting credit for: being an industry standard and being (somewhat) easier to use.

While component models were busy finding their place in the industry, a parallel effort was taking place. The W3C established itself as the organization dedicated to making the Internet technology, and the Internet became recognized as a part of one's infrastructure, causing an intersection of these two endeavors. So is this code reuse strategy that much different from its predecessors?

The Differences – A Linear Trend

One of the points of confusion about Web services is not so much what they are (read the specs), but rather when to use them. Technically, Web service technology can be used to facilitate every level of interoperability, from a single line of code to an entire ASP application, but that would be violating the most popular adage: use the right tool for the job. Much of how Web services fits into the bigger picture can be derived from examining the trends.

- ***Larger level of granularity:*** One thing that objects, components, and services have in common is that they're all reusable blocks of code. But one clear trend is that the size of the block is growing. Applications (especially e-business applications) are larger and stretch across more pockets of functionality, and middleware is evolving to support that. Typically, components are larger than objects, and services are larger than components. Often the effort put into understanding someone else's code is costlier than rebuilding it, but as the size of the reusable block of code grows, the trade-off balance shifts.
- ***Reduced effort required to reuse code:*** C++ users rely on inline comments, old design specs, or lunchroom chatter to effectively share objects, but Web services have help for this step built in. Besides standard interfaces, there's UDDI and WSLD to find and discover services. Services are striving to reach the nirvana of productive cooperation between strangers. Perhaps we've crossed the threshold where reusing code requires less of an investment than writing it.
- ***Breadth of scope:*** With each generation of reusable objects, the reach extends farther and farther. This is in part due to the size and ability to access more and more diverse assets, but services makes a large jump by being able to cross the Internet and by being an industry standard.
- ***Increased focus on information:*** In today's interaction of services, the payload gets the spotlight. Sending business documents and sharing information is often the focus of e-business applications. Until the invention of XML, data sharing required both the sender and the

Continued on page 8

CJAENICKE@MEDIAONE.NET

AUTHOR BIO

Coco Jaenicke was, until recently, the XML evangelist and director of product marketing for eXcelon Corporation. She played a key role in the successful development and introduction of eXcelon, the industry's first application development environment for building and deploying e-business applications. Coco is currently an independent consultant.

Letters to the Editor

Applause for XMLEdge

Just wanted to say thanks for the hospitality at the show. I was pleasantly surprised by the warm reception I received from everyone at **SYS-CON**.

Many of the XMLEdge sessions were worthwhile and practical. As an IT manager, I learned of the potential development cost savings of next-generation, XML-based, browser UI technologies.

It was also great to see the OASIS organization promoting its ebXML standards for companies wishing to do business with one another. Companies big or small looking to implement supply-chain or B2B integration with their vendors and partners should see the immediate benefits of the ebXML standards.

Herb Ramos
via e-mail

Plea for Help

I've been reading ***XML-J*** with interest. I manage 90,000 film and video titles. Several years ago I taught myself a little XML and built an online catalog/metadata generator. At a base level, it spits out Qualified Dublin Core Metadata expressed in XML and (next week – I hope) RDF. It uses XSLT to produce a token back-of-house HTML view (needs design).

There is a range of further developments that I need quite urgently. These include front-end design work and some back-end applications. Our IT manager does not think XML provides a robust application development environment. He wants to take the catalog and its development off-line and develop it on what he calls a "serious" object-oriented development platform (Oracle). I've been quoting arguments that SOAP, WSDL, and UDDI, specifically, form the backbone of tomorrow's Web-centric infrastructure.

Would you point me to an article that might support my argument?

Simon Pockley
via e-mail

It looks like you have a problem faced by several folks who are trying to adopt XML as a viable technology for their e-business applications. For one thing, you want to refer to SOAP, WSDL, and UDDI only if you're planning to develop your catalog in the form of Web services. Web services are a fairly new paradigm, and it's hard to find real-world examples in terms of existing applications.

A better approach would be to discuss with your manager the momentum behind Web services that's created by companies like Microsoft, Sun, IBM, Oracle, SAP, and BEA. If you research the work being done by these companies in XML and Web services, and the breadth of tools they offer, you should be able to make a good case for yourself.

You may want to make a case for back-end integration using XML as a standard approach if you're ultimately going to go to a variety of legacy systems. In such cases loose coupling using SOAP messages makes sense. However, if you're working with just one system, it's often hard to justify the need for an open standard. So the selection needs to take into account whether you're designing for the future or a one -shot deal.

It's easier to make the case for the front end because XML formats can be used in a variety of presentations and can thus apply to a variety of presentation devices, including small devices. Again, the applicability depends on your environment. If you have a single browser front, it may be more difficult to justify.

However, using Oracle and XML aren't necessarily distinct. Oracle has support for XML. So do all the other database vendors. However, if you can achieve your purpose using plain database queries, XML will be overkill.

I don't know of any particular article that will help. However, OASIS has a specification for a catalog at www.oasis-open.org/committees/entity/spec.html*. And the following books may help:* XML for Managers *by Kevin Dick and* The XML Handbook *by Charles Goldfarb.*

Letters may be edited for grammar and clarity as well as length.

Please e-mail any comments to Ajit Sagar at **ajit@sys-con.com**.

Industry Commentary *continued from page 7*

receiver to know exactly what was to be transferred and how to read and process it. This had to be hard-wired into the application, greatly reducing flexibility. Self-describing XML fundamentally changes how services can share information and has taken code reuse to the next level.

Because all these changes are linear, one could argue that nothing is truly new since the days of the first object-oriented programming language (and even that could be seen as a logical derivation from earlier technology, but I don't want to date myself). In fact, Web services are really just the original vision with more of the bugs worked out. On the other hand, some of the steps that have been taken are giant steps.

CORBA and COM: Could It Have Been You? Perception vs Reality

So, could some of the component models have evolved to capture the glory that swirls around Web services? Technically, they could have added support for XML and the ability to cross the firewall, and many have. But there are two other problems.

First is the issue of being proprietary versus being an industry standard. The only difference between the two, of course, is the number of people sharing the opinion, so this is more perception than reality. The other is ease of use. While Web services are certainly easier to use than traditional components, they're not as easy to use as HTML. Since Web services standards are so tightly associated with HTML and the W3C, the perception gives services an extra edge.

Looking Forward

Will these trends continue? Absolutely. Apps will continue to grow and reuse will continue to be simplified, all driven by businesses' need to be agile. Many of the known "bugs" in the system, like security, monitoring, billing, and workflow, will be tackled and solved in time. The trajectory, however, will remain the same: easier and more productive reuse of existing assets.

XML AND

MARK.ONEILL@VORDEL.COM

Mark O'Neill, chief technology officer of Vordel, oversees the development of Vordel's technical strategy and product development in the areas of XML and public key cryptography. Mark designed and implemented Internet EDI solutions for Ireland's largest EDI value-added network, Eirtrade Services Ltd. He holds a double-honors degree in math and psychology from Trinity College Dublin and studied neural network modeling at Oxford.

THE XML AND SECURITY WORLDS ARE COMING TOGETHER IN INTERESTING WAYS

SECURITY

WRITTEN BY MARK O'NEILL

As XML becomes the de facto format for businesses to communicate over the Internet, the need for security comes to the fore. Digital security has always been about the compromise between convenience and peace of mind. This holds true for XML also.

The proposed advantages of XML for digital commerce – the opening up of internal systems to trading partners via commonly agreed standards – are also concerns for security. These concerns are now being addressed by a number of industry initiatives.

This article describes a selection of these initiatives: the W3C's recent XML Signature specification and its relationship to SOAP, the OASIS SAML (Security Assertions Markup Language) initiative, and XKMS (XML Key Management Specification). Together, these initiatives are setting in place the infrastructure that will allow XML to travel safely between enterprises.

History

A common thread in the debate about XML and security has focused on whether or not to put the security layer within the XML document. Some of the early non-XML B2B integration frameworks, such as OBI (Open Buying on the Internet), which began in June 1997, incorporated X.509 and digital certificates and digital signatures at field level into their document sets. Then the early XML-based B2B integration frameworks, such as Open Trading Protocol (OTP), followed suit with security-specific tags. At this point opinion shifted and it was thought best not to mix up the XML data payload with security and authentication information. As the HTTP POST protocol became the commonly accepted method of transmitting XML, it was felt that SSL should be used since it comes "for free" with HTTP. XML can be transmitted just as well over an SSL connection as over a plain HTTP connection, albeit somewhat more slowly. The first SOAP draft (1999) avoided the authentication question, deferring it to later drafts, and suggested the use of SSL. However, although SSL handles authentication, it doesn't address digital signatures. The W3C then became involved, setting up the XML Signature Working Group to produce the XML Digital Signature Specification. XML-DSIG is an important standard because it supports the digital signing of any digital content, not just XML. Thus the debate has come full circle; the signature is now once again part of the XML document, except that now the signature format is a common standard that can be archived and interpreted by any piece of standards-compliant software.

XML Signature

The XML Signature standard describes a set of XML elements and attributes that are used to store information about the hashing and encryption algorithms used to generate a digital signature, as well as, of course, the signature itself. In addition, the public key used to verify the signature can be incorporated within the <Signature> block; alternatively, the address of the public key directory that includes the public key can be included.

The discussion relating to the design of the XML Signature standard yielded a number of interesting questions. Some touch on philosophical issues and get to the core of the concepts behind structured data and its representation onscreen. The XML Signature standard mandates that only what is "seen" should be signed. The word *seen* is in quotation marks because the user may perceive the information in another media other than visual – through sound, for example.

It's important to secure the actual data that was presented to the person. This means that if XML is rendered onscreen using a stylesheet, then the visual representation of the data must be signed since this is what the user actually sees. It's been suggested that the components used to render the XML should also be signed – the XML Signature specification says that the data must be signed along with "whatever filters, style sheets, client profile or other information that affects its presentation." These items may include the browser itself, even video drivers or font packs, or ultimately the operating system itself. The important point is that the user's decision to sign is based on the visual representation of the XML data, not the underlying XML itself.

The Identrus PKI group – a consortium of banks that issue digital certificates signed by the Identrus root certificate authority (CA) – requires that users be presented with a bitmapped image of the document that is to be signed. This bitmap isn't useful for subsequent data processing but instead serves as a record of what the user saw. The signing software must ensure that the document that the user views isn't obscured by another application in the foreground. Identrus uses an XML format, CSC (Certificate Status Check), to authenticate users.

Another interesting aspect of XML Signature is that the document itself must be protected so no changes happening to it in transit can invalidate the signature. To understand why this is important, you have to understand what a *hash* is. A hash is a value produced by a one-way mathematical function run on a piece of data. If others run the same hash function onto the data, they obtain the same hash value.

This is how signatures work – this hash value is encrypted with the private key of the signer, and then anyone with access to their public key can decrypt the original hash, compute a new hash based on the data they have received, and make sure that the two hash values are the same. XML presents a number of problems for hashing, however. An XML document may contain some white space between tags, for example, and this white space may be lost when a DOM or SAX processes the XML. Similarly, the order in which tags or attributes occur in an XML document may be changed when it's loaded into a DOM or SAX processor.

The problem with this scenario is that when the application computes a hash of the document, the white space or the tag order having changed, the hash won't match the original hash, so the signature won't compute. In addition, certain differences between file formats on different operating systems can cause XML documents to subtly change as they are sent between disparate machines. These issues are to be solved by *XML Canonicalization*. XML Canonicalization defines a standard way to normalize XML information between operating systems. So-called *canonical* XML is intended to be platform neutral.

An example of an XML Signature is shown in Listing 1. The SignatureMethod tag tells us that a combination of RSA (for public key encryption) and SHA-1 (for hashing) was used to create this signature. The X.509 certificate used to verify the signature is included with the signature itself. This signature is appended to the document it signs.

PKI – Binding a Key to a Person

The XML Signature standard specifies XML digital signature processing rules and syntax that prove that a document was signed using a certain private key; a Public Key Infrastructure (PKI) then binds that key to a user's identity. Note that there are two clauses in the previous sentence. Digital signature algorithms provide the mathematical proof of a transaction. However, unless the private key is linked to a person or organization, that proof is just mathematical. PKI is used to link the transaction to the person, making use of publicly available directories to store the public keys used to check the digital signatures and referencing a security policy document to enforce identity checks on applicants for digital certificates. Implementing a PKI can be a notoriously difficult and expensive undertaking, so many organizations rely on global PKI services such as VeriSign or PricewaterhouseCoopers beTRUSTed.

As we've seen, PKI brings a lot of value to XML. Conversely, however, with the arrival on the scene of XKMS (XML Key Management Specification), the world of PKI is beginning to become XML-enabled. XKMS is proposed by VeriSign, Microsoft, and webMethods, and has been submitted as a W3C note. It comprises two parts: the XML Key Information Service Specification (X-KISS) and the XML Key Registration Service Specification (X-KRSS).

X-KISS allows a client application to delegate part or all of the tasks required to process an XML Signature to a trust service. This is useful for developers who don't want to implement the signature checking themselves, or who want to delegate this functionality to an application service provider that may be optimized for signature checking (e.g., through hardware acceleration).

X-KRSS is an XML-based replacement for existing PKI file formats that are used when a user applies for a digital certificate. XML brings the same advantages to PKI as it brings to other industries – open standards, platform independence, and human readability.

XKMS looks likely to take off, not least because Microsoft is bundling it into its .NET initiative.

Web Services – Component-Based Computing Takes to the Web

The long-standing drive toward component-based computing in IT architectures is now moving to the Web. Components that are physically located on different computers can run together as one solution, using technologies such as SOAP (for enveloping XML on the wire), UDDI (for publishing information about available services), and DSML (for accessing directories over the Web) over frameworks such as .NET, Jini, or E-Speak.

A simple example of a Web service is a stock quotation object that can be instantiated over the Internet by an application that requires such a tax-calculation feature. By tying together Web services, "business Webs" – dynamic collections of businesses – can be spawned on a massive scale. An example of a business Web is a retail store that uses UDDI to publish its online catalog; the catalog can then call the company's shopping cart and a third company's credit card transaction service. Development tools such as Microsoft's Visual Studio.NET and Bowstreet's jUDDI allow developers to link Web services to create business Webs, often without any need for programming.

SOAP is firmly established as the enveloping protocol of choice for Web services. Until recently, SOAP did not address the requirement for security. But in January 2001 Microsoft and IBM proposed in a W3C note the integration of XML Signatures into the SOAP 1.1 Envelope via a new <SOAP-SEC:Signature> header entry. The various Web services frameworks – .Net, Jini, and E-Speak – will most likely use XML Signature-enabled SOAP messages.

E-Speak is something of a special case because it was the first fully operational Web services design, initially announced by Hewlett-Packard back in 1999, and was recently updated to comply with the SOAP specification. Certificate-based security is included in E-Speak in the form of fine-grained, rule-based security that uses attribute certificates. It remains to be seen if SOAP-level security will supplant this.

The advent of Web services opens up some important questions for security. If it's so easy to string Web services together to create a business Web, then what's to stop a hacker from exploiting this? What is needed is a way of certifying Web services. Otherwise a Web agent that searches for services has no way of knowing what services to trust. Centralized, trusted, UDDI directories are one way of answering this security question. However, it remains to be seen how well this option will scale. The other option would be to use a certification system similar to Microsoft's AuthentiCode, where the onus is on vendors to register and sign their service. This has the advantage of retaining the peer-to-peer nature of the Internet, but still depends on the existence of a service to check credentials. As we've seen, XKMS fits the bill as a protocol to deliver this.

And What About Firewalls?

One very special reason why XML-specific security is important is that Web services typically use the Web ports, thereby bypassing firewall restrictions. An example of this trend is SOAP, which earlier in its lineage used to travel over port 135 (the RPC Endpoint Mapper port), a port that is typically blocked by firewalls for security reasons. Now SOAP uses the Web ports and so avoids firewalls.

Other examples are the new XML interface on Microsoft SQL Server 2000, or the XSQL feature that allows Oracle 9*i* to conveniently read in a stream of XML. For an IT manager it's an appealing prospect to Internet-enable an application by opening it over Web ports via an XML interface. Quite often the fact that an application is blocked by a firewall appears to users as if the application "just plain doesn't work." Users typically don't understand that a protocol is blocked by the firewall for a reason.

This problem held up the spread of CORBA, even resulting in some CORBA vendors resorting to writing their own firewalls. PKI rollouts have also been affected by this problem, which results in essential LDAP directory lookups (which use port 389) being blocked – hence the need for Directory Services Markup Language (DSML, pronounced *dismal*) to provide an XML-based directory lookup over the Web ports.

The XML-based Internet does away with the possibility of denying network traffic based on specifying TCP/IP port numbers. Next-generation firewalls must be capable of dipping into XML streams traveling over Web ports to check their payloads, much like today's e-mail virus checkers dip into e-mail data streams on mail servers. In the case of XML signatures this authentication can be done locally or by sending the signature block to an XKMS trust service. However, if the XML

	Secure Transports: SSL and/or HTTPS	Messaging Protocol: S/MIME and JMS	Enveloping Formats: SOAP, etc.	B2B Application Protocol: ebXML, BizTalk, etc.
XML Signature (W3C & IETF)	Independent of transport protocol	Independent of messaging protocol	Proposal for inclusion in SOAP header	BizTalk Framework 2.0 includes XML Signature support
XKMS (submitted to W3C)	Independent of transport protocol	Independent of messaging protocol	Uses SOAP for enveloping	Proposed support in .NET
SAML (OASIS)	Independent of transport protocol	Independent of messaging protocol	Bound to SOAP	N/A

TABLE 1 XML Security standards vs B2B network layers

stream is encrypted, a traditional firewall is of limited use because it simply can't read the data. SOAP partially gets around this problem by allowing the SOAPAction method name in the HTTP header to travel in the clear so that a firewall can route the document. But this has the disadvantage of giving away information about which Web service is being accessed.

The SOAP specification includes the SOAP-specific M-POST command that enables SOAP-compliant programs to add header information to the HTTP protocol to allow fine-grained, rule-based filtering and handling of SOAP messages by firewalls and proxy servers. This of course relies on proxies and firewalls being configured to recognize M-POST.

S2ML and AuthXML – Two Become One?

In November 2000 two separate initiatives were announced to develop an XML standard for transporting security information between online commerce systems. The two initiatives are S2ML (Security Services Markup Language), led by Netegrity, and AuthXML, led by Securant Technologies. The goal of both initiatives is to implement Single Sign-On, one of the holy grails of computing, between online trading environments.

This service is needed because online commerce typically involves more than one Web site or Web service, and these may need to share information about a user. S2ML or AuthXML would assist partners and affiliates in linking their exchanges to share "entitlement" information – for example, credit limits and "gold card"–type profiles. Also, both protocols would eliminate the need for users to repeatedly enter registration information onto multiple Web sites. Participants in S2ML include webMethods, Sun, VeriSign, and Jamcracker. In addition, the ebXML working group has endorsed S2ML. Participants in AuthXML include Check Point, Novell, and ValiCert. Some vendors, like some Florida voters, signed up to support both competing initiatives.

In view of the fact that S2ML and AuthXML address the same requirements but aren't interoperable, OASIS (part of the Open Group) set up a technical committee for XML-based security services to merge the two initiatives into a single standard. It was felt that a single standard would be a more favorable outcome for the industry than two competing initatives. After all, by definition a "standard" should be something that everyone uses. The OASIS initiative to merge AuthXML and S2ML is still in the early stages, having started in December 2000, but is gathering momentum. The initiative has been christened SAML – Security Assertions Markup Language.

How This All Fits Together

The initiatives described in this article fit into various parts of the four layers shown in Table 1. It's expected that XML Signature will be incorporated into many of the B2B integration frameworks, via the proposed SOAP XML-DSIG header extensions.

The next few months should be interesting for both the XML and the security worlds because they're coming together in interesting ways. XKMS is bringing the XML message of common standards to the digital security industry, notorious for its fragmented standards. Similarly, initiatives such as XML Signature and OASIS SAML are bringing the vital level of trust to business-to-business trading on the Internet. These events should lay the secure foundations for the much-anticipated growth of business Webs.

LISTING 1 XML Signature Example

```
<?xml version="1.0" ?>
<Signature Id="Vordel"
      xmlns="http://www.w3.org/2000/CR-xml-c14n-20001026">
<SignedInfo>
<CanonicalizationMethod Algorithm="None" />
<SignatureMethod Algorithm=
   "http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
<Reference URI=
   "http://www.w3.org/TR/2000/CR-xml-c14n-20001026">
<Transforms>
<Transform Algorithm=
   "http://www.w3.org/2000/CR-xml-c14n-20001026" />
</Transforms>
<DigestMethod Algorithm=
   "http://www.w3.org/2000/09/xmldsig#sha1" />
<DigestValue>qyd5dHCHsQ1GXw0j6hk6PZtF8vE=</DigestValue>
</Reference>
</SignedInfo>
<SignatureValue>
NrZOJ7rEyIPmLs/CoK2gQJ32EWwkTnAkhuzUMrjs/+WwJdJ+3XoP
</SignatureValue>
<KeyInfo>
<X509Data><X509IssuerSerial><X509IssuerName>
c=IE, o=Vordel, ou=TS, cn=VordelCA, mail=info@vordel.com
</X509IssuerName>
<X509SerialNumber>970241782</X509SerialNumber>
</X509IssuerSerial>
<X509SubjectName>
c=IE, o=Vordel, ou=Dev, cn=Mark, email=mark@vordel.com,
telephoneNumber=35312153333
</X509SubjectName>
<X509Certificate>
-----BEGIN CERTIFICATE-----
MIIC2TCCAcGgAwIBAgIEOdS29jANBgYkCgYEAr2emzUvznx9/j
eFNc5NUImceS9x9QSP63cxkwlGAQYS3OkOFShmeF6xvt8ra2Ui
wS0xO1FYXQu7mRIAKQe9zhQaIP63NlsqfuRJLNkRFkHstfZtTlE
SzAe5LosLGVgeU8ocT+8f6zu3LkcgqfWJhxq79YScl9OixBYD6jA
IIDC4IHgEOyDCLhKaJgZ2eAnepx4Mk+fSPmGvN7uDhUIk/OujQ
OwlOG4qYzrd4d4Vax/QV6GXn2UpT894h0giEBxZczY4xIkCsdXI
GF+PGIfcq1WdPYMG+Nvz661QMrTxGYiG8Aws2R8+29mw6JkY
TzcpItNw95FQoM1MpeMfUZcm/Ja7Fon2Qfp9oeGTlNE+Q==mk
l213blktrh[-qwDFSgfEWTet=2tewgfsm,qWuGrt
-----END CERTIFICATE----
</X509Certificate>
</X509Data>
</KeyInfo>
<Object>
<SignatureProperties>
<SignatureProperty Id="TimeStamp" Target="#Vordel">
<timestamp><date>2001Apr02</date>
<time>10:08:04</time></timestamp>
</SignatureProperty>
</SignatureProperties>
<DirectoryIPAddress />
</Object>
</Signature>
```

DOWNLOAD THE CODE @
www.sys-con.com/xml

[WRITTEN BY SIMEON SIMEONOV

My world is my own creation

PrivateUDDIRegistries

The last issue of **XML in Transit** focused on the categorization and classification capabilities of UDDI. This month we look at the means for customizing the power and flexibility of UDDI for your business through the use of private UDDI registries. Why would a company host a private UDDI registry?

UDDI provides a standard message API to publish and find service descriptions from a repository. This standard is supported and used by many organizations; there are multiple UDDI registry implementations to choose from. Some organizations advertise all of their Web services descriptions in the UDDI Business Registry. These companies also use the registry to search for any business partner's Web service descriptions.

The UDDI Business Registry is in a well-known location (www.uddi.org) and therefore highly visible on the Web. Everyone in the Web services community knows about UDDI, and knows how to use the UDDI Business Registry for find and publish. Advertising in the registry maximizes the visibility of a business and the services it makes available to potential business partners.

The UDDI Business Registry isn't the only place to register business and service descriptions in a UDDI format. Many organizations are choosing to host their own private UDDI registries. These companies make this choice for several reasons:

- Control of access to the information
- Control over updating the information
- Reliability of the content of the registry

Hosting a private UDDI registry doesn't preclude using the UDDI Business Registry. Far from it. Many organizations coordinate use of their private UDDI registries with occasional access to the UDDI Business Registry.

The broad visibility of the UDDI Business Registry has a downside for some organizations that want to restrict who is allowed to view sensitive service description information and the network address that accesses their Web services. For this reason many organizations advertise just their company in the UDDI Business Registry; the only Web service they advertise is the UDDI Inquiry API Web service to one of their private UDDI registries. In this way potential partners who are interested in learning more about a business's Web services capabilities are encouraged to contact the business's private UDDI registry. The business hosting the private UDDI gains control over access to Web services information through a registration and authentication scheme. Because they control their private UDDI registry, access can be tracked and monitored, and, if necessary, follow-up can be initiated with a potential partner showing interest in the business by executing a find operation.

Some organizations use a private UDDI registry to control the visibility of service description information. Some organizations simply aren't comfortable with having this information managed by another organization, including an open consortium such as UDDI.org.

For Web services with a network location that changes with some frequency, direct control over changes to the service description entries in UDDI is necessary.

Many organizations host a private UDDI to ensure consistency of service description information to support runtime discovery of Web services. This consistency is at the business level (who the partner is) and determines how the service is described (use of service interface definition standards).

The UDDI Business Registry contains business and service information about companies from a broad range of industries. Few of these entries are of interest to any given businessperson searching for a particular partner or Web service implementation. For those businesses categorized within a particular industry (using the North American Industry Classification System taxonomy, for example), not all of these businesses are desired partners. Worse, there's no guarantee that a company categorized in a particular way does in fact do business in that industry. Some organizations use a private UDDI to ensure that only services from approved business partners appear in the services registry. As new partners are approved, their UDDI entries are added to the private UDDI registry. As business relationships dissolve, the entries for those partners are removed.

The flexibility provided by UDDI to register services regardless of how they're described is both a benefit and drawback for users. For example, most organizations have enabled their applications to consume Web services of a given type. If a Web service isn't based on one of a preselected set of Web service types, then it isn't useful to the business's applications and therefore shouldn't be registered in their private UDDI registry.

A private UDDI registry thus offers a target-rich environment so that an application, at runtime, can do a find against the registry. A business policy, defined by human developers at design time, is used as the search pattern in the find operation.

This article is based on the UDDI chapter in *Building Web Services*, to be released this month. It appears here in slightly different form by permission of the publisher, Sams. Contributors to the book are Doug Davis, Steve Graham, Yuichi Nakamura, and Ryo Neyama from IBM; Toufic Boubez from Saffron Technology; and Glen Daniels from Macromedia.

AUTHOR BIO

Simeon Simeonov, chief architect at Macromedia, Inc., is a member of the W3C working group on XML protocol and the J2EE expert groups on XML business messaging and XML data binding.

The pattern describes the desired Web service type and other characteristics, such as nonfunctional requirements specific to the business policy. The Web services discovered by these search criteria will fit the business need of the application, be in a format that is directly consumable by the application, and will be hosted by a known and approved business partner.

Supporting dynamic runtime discovery and binding to Web services at runtime is one of the key points of flexibility in a service-oriented architecture. This flexibility is important for loosely coupled application integration, either within an organization or between business partners.

Let's examine five types of private UDDI nodes and how businesses might use them. For example purposes I'll use three fictitious companies: SkatesTown (a skateboard manufacturer), e-Taurus (a small wheels and bearings e-marketplace), and WeMakeIt Inc. (a participating company in the e-Taurus marketplace).

Five Types of Private UDDI

The UDDI specification was defined to allow the possibility of private UDDI registries. A private UDDI registry can implement some or all of the UDDI APIs. A private UDDI is certainly not bound to any restrictions articulated by the UDDI operator's agreement, and in particular does not participate in the replication mechanism within the UDDI Business Registry. A private UDDI registry may alter the behavior of the common UDDI operations, for example, requiring an authentication SOAP:header on the find operations. A private UDDI registry can offer additional APIs over and above those defined by the UDDI specification.

The five broad categories of private UDDI use summarized here are described in detail below:

1. E-marketplace UDDI
2. Portal UDDI
3. Partner catalog UDDI (also known as Vetted partners or Rolodex-like UDDI)
4. Internal Enterprise Application Integration UDDI
5. Test Bed UDDI

E-Marketplace UDDI

This type of private UDDI node would be hosted by an e-marketplace, an industry standards organization, or some other consortium of organizations that compete/participate in an industry. All publish and find (inquiry) APIs are typically deployed for Internet access.

The e-Taurus organization hosts an e-marketplace private UDDI on its Web site. All buyers and sellers of small wheels, nylon, steel and aluminum, and related components, such as wheel bearings, come to this site and use this private UDDI registry.

The entries in the e-marketplace type of private UDDI are all related to businesses within a particular industry or a narrow range of related industries. Further, the membership process allows entries in this UDDI to be prefiltered to include only legitimate businesses participating in the industry. The membership process can also restrict who is allowed to invoke find operations against the UDDI node. Private e-marketplace UDDIs can implement custom taxonomies and the associated validation processes as discussed in the previous **XML in Transit** (***XML-J***, Vol. 2, issue 11).

> "A private UDDI registry offers a target-rich environment so that an application, at runtime, can do a find against the registry"

The e-Taurus organization provides a mechanism for registration at its Web site. Each user from each participating member organization must register to receive an authentication token. The token must be included as a SOAP:header on any find operation and is required by the UDDI API specification to be part of any publish operation issued against e-Taurus's private e-marketplace UDDI. The token allows e-Taurus to track which members are doing publish and find operations and support a small subscription fee based on the size and number of publish operations done by each company as well as the number of find operations and the size of the result sets.

An e-marketplace UDDI is a rich environment for finding Web services metadata for doing business within a particular e-marketplace or industry. The e-marketplace hosted by e-Taurus is clearly the place to be seen in the small wheel and bearing world. And the e-marketplace UDDI is the logical place to find industry-specific custom taxonomies (standard product coding hierarchies, NAICS categories, etc.) as well as standard Web service interface definition tModels for common business processes in the industry.

This type of private UDDI allows an e-marketplace organization to provide value-add to Web services advertisement and searching; for example, Quality of Service monitoring on a partner's Web services response times and Better Business Bureau–style industry self-monitoring of business practices of its members' Web services.

An e-marketplace type of private UDDI registry is where finds in serious B2B applications happen. Because e-Taurus monitors the participants in the e-marketplace, SkatesTown can trust that the entries in the e-Taurus UDDI registry are all legitimate suppliers. Further, e-Taurus can provide value-add features to a UDDI search at the business level. For example, its private UDDI registry can sort the result set of a particular kind of find operation for a Request for Quotes (RFQ) service by price value by automatically checking the RFQ against suppliers' catalogs.

Portal UDDI

Web services technology is defining the standard way to use the Internet for B2B applications. This use of the Internet can be contrasted with the current eyeball or HTML-based World Wide Web by calling it the *semantic* Web or the *transactional* Web. Just as a company has a Web presence on the eyeball Web (www.WeMakeIt.com), so too might it have a presence on the semantic Web (e.g., www.WeMakeIt.com/services/uddi/). The private UDDI representing the organization's semantic Web presence is called a *portal UDDI.*

The portal UDDI hosted by WeMakeIt Inc. resides within the company's demilitarized zone. The entries in WeMakeIt Inc.'s private portal UDDI registry contain descriptions for those Web services that WeMakeIt Inc. wishes to provide to external partners. Clearly it is in the company's interest to keep the find APIs available on the Internet; however, WeMakeIt Inc. doesn't allow access to the publish APIs from the Internet, restricting publish to internal processes only. WeMakeIt Inc. also hosts its product code taxonomy in its private portal UDDI registry. Partners that wish to do business with WeMakeIt Inc. examine that registry to determine what formats the company accepts for purchase orders and the WSDL description of the purchase order placement service, including its network address.

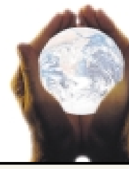
When WeMakeIt Inc. wants to deploy a new Web service, it publishes the Web service's description in its portal UDDI.

As part of WeMakeIt Inc.'s businessEntity registration in the UDDI Business Registry, the URL for its private portal UDDI registry is used as a discoveryURL element, with the useType attribute set to urn:uddi-inquiry-api. A segment of WeMakeIt Inc.'s businessEntity entry is shown in Listing 1.

The portal type of private UDDI registry gives a company ultimate control over how metadata describing its Web services is used. For example, companies are free to restrict find access to the registry. Companies are able to monitor and manage the number of find operations being made against their data and potentially derive information about the interested parties.

Partner Catalog UDDI

This type of private UDDI node sits behind the firewall. It too provides a target-rich environment against which Web services finds and binds can be made. A partner catalog UDDI registry contains only Web service description metadata published by trusted business partners. WeMakeIt Inc. has a partner catalog UDDI registry containing entries for those organizations it has formal business relationships with. In most cases neither the publish nor the find APIs to the partner catalog UDDI registry are available over the Internet, restricting access to all APIs to internal applications only.



Businesses today do business with organizations they know. Use of a partner catalog UDDI registry allows an organization to build applications in a service-oriented way, taking advantage of dynamic binding against Web services at runtime based on a Web service interface built into the application at design time. Because this kind of registry contains only approved business partners, this style of dynamic binding doesn't imply the risk of dealing with an unknown service provider. No matter which Web service is returned by the find operation, we know it's provided by a validated business partner.

LISTING 1 Customized use of discoveryURL

```
<businessEntity authorizedName="…"
  businessKey="EF433600-4C7D-11D5-9B9D-
  A45413861BC0" …>
   <discoveryURLs>
      <discoveryURL useType="businessEntity">
         http://www.someoperator.com/uddiget?
          businessKey=EF433600…
      </discoveryURL>
      <discoveryURL useType="urn:uddi-inquiry-api">
         http:// www.WeMakeIt.com/services/
          uddi/servlet/uddi
      </discoveryURL>
   </discoveryURLs>
   <name>WeMakeIt Inc.</name>
...
</businessEntity>
```

DOWNLOAD THE CODE @ www.sys-con.com/xml

WeMakeIt Inc. uses its partner catalog private UDDI to help in its own supply chain automation systems. Web services technology allows WeMakeIt Inc. to do process integration with their suppliers. Using Web services, WeMakeIt Inc. can reduce transaction costs for supply chain tasks, such as supply reordering, in a way that doesn't lock them into any particular supplier.

The management at WeMakeIt Inc. agrees to do business with some supplier. Someone on the business development automation team at WeMakeIt – let's call her Joanna Pravard – examines the UDDI entries for that partner (either from the UDDI Business Registry, an e-marketplace UDDI like e-Taurus,or the supplier's portal UDDI. Joanna copies these entries into WeMakeIt Inc.'s partner catalog UDDI.

This process is repeated for each supplier as new suppliers are found and business terms negotiated. The set of suppliers changes over time as business relationships are formed and dissolved. Changes to this set are reflected by changes in the entries within WeMakeIt Inc.'s partner catalog UDDI registry.

To make this dynamic binding application support complete, Joanna restricts the kinds of entries that can be published into the partner catalog UDDI registry. Joanna works with each supplier to make sure the supplier's businessServices are properly categorized according to WeMakeIt Inc.'s product code taxonomy. She also makes sure that each tModel referenced by these businessServices is from a set of approved, standard tModels supported by WeMakeIt Inc.'s applications. This allows Joanna to guarantee the shape of entries that are placed within the UDDI node and therefore what applications can expect in response to find operations.

Internal Enterprise Application Integration UDDI

The Internal Enterprise Application Integration type of UDDI registry is similar to the partner catalog type except that it contains entries for Web services provided by other departments or groups within an organization. Many organizations treat their partner catalog UDDI and Internal Enterprise Application Integration UDDI as logical views on the same physical registry.

The major difference between the Internal Enterprise Application Integration type of private UDDI and the partner catalog type is the potential for a common administrative domain that can dictate standards (which tModels are used, common use of WSDL portTypes, etc.). This allows the Internal Enterprise Application Integration type of UDDI registry to operate with different publish restrictions than those suggested for the partner catalog type. For example, it could restrict the publication of new tModels and thereby restrict publishing of businessService entries and bindingTemplate entries to accept only entries associated with a fixed set of tModels. These tModels correspond to the technology standards chosen by the decision makers controlling the common administrative domain.

Of course, this kind of UDDI registry exists completely hidden behind the organization's firewall. Publish and find operations are restricted to applications within the organization.

Test UDDI

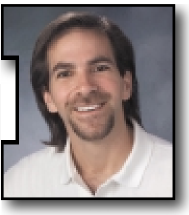
Programmers use this type of private UDDI registry to test applications. The testing can be for both requestor applications and provider Web services.

SkatesTown uses this type of private UDDI registry to ascertain that the UDDI entries describing its purchase order placement Web service are accurate and that UDDI-aware tools can generate proxies from the UDDI entries to access its purchase order placement Web service.

WeMakeIt Inc. also uses a test UDDI to determine its application's ability to cope with external services. For example, Joanna Pravard uses a test UDDI to make sure that different permutations of the RFQ Web service provided by various suppliers all run correctly with WeMakeIt Inc.'s reorder application. Joanna runs trials against any new UDDI entry discovered in the UDDI Business Registry, or an e-marketplace's private UDDI, or a supplier's private portal UDDI. First she copies a UDDI entry from the source UDDI registry to the test UDDI, then she runs a battery of tests to make sure WeMakeIt Inc.'s applications can use the information found in the entry. Then, only after testing, she promotes the entry to the WeMakeIt Inc.'s partner catalog UDDI.

SIMEONS@MACROMEDIA.COM

[WRITTEN BY DAVID SILVERLIGHT



The Silverlight technique – or Ode to Steve Muench

Shedding a Little Light on XML

Wouldn't it be great to have something named after you? Up to now I've had no luck. Off the top of my head, the only way for this to happen is to (1) have children, (2) invent a lifesaving technique like the Heimlich maneuver, (3) die of a painful disease (Lou Gehrig's disease, for example),…

…(4) name a star after yourself (the "XML Star" was still available the last time I checked), (5) invent a clever method of grouping XML data (aka the Muenchian Method).

Hopefully I will never qualify for option 3. Steve Muench, however, has managed to gain immortality by developing a very innovative way to group XML data.

Grouping XML data is a common programming dilemma, which will be resolved in this month's column using his technique, that is, the Muenchian Method. The Muenchian Method is not inherently obvious, primarily because it employs two functions, key() and generate-id(), neither of which is widely used. However, once they're demonstrated as part of a solution, their usefulness will stand out.

Black Belt

What Is the Muenchian Method?

The Muenchian Method is a novel technique used for grouping XML data using the XSLT key() function. The technique offers a clever solution to the questions "How do I group unsorted XML data efficiently?" and "How do I find the unique values in a set of values?"

While the method is used for grouping data, it's also used to find the unique values in a set of values. It has widespread applications beyond grouping. For example, you might want to count the number of states represented by a list of salesmen's territories. Though it may not sound like a grouping problem (in fact, it *isn't* a grouping problem), you'd still use the key tables in the Muenchian Method to obtain the answer.

To further explain this technique, I'll use a sample XML document containing unsorted product information. The output generated will produce a list of products grouped by region, then by product name. Both groupings will be sorted.

Overview: A 50,000-Ft View of the Algorithm

Two loops will be used to process the product information.

The outer loop will iterate through all unique "regions" in the XML document. Each iteration of the loop will display the region name, sorted alphabetically, as the header for the HTML table that's generated. At this point the key() and generate-id() functions will be integral in generating a distinct set of regions.

The inner loop will iterate through all products for the current region and will display product information alphabetically. Here, the key() function will be integral in allowing us to quickly access the products for a given region.

Breakdown

***Step 1:* Define the primary key to be used in the Muenchian grouping.**

Defining the primary key is the first step in Muenchian grouping. Using the xsl:key element sets up the groundwork for later accessibility to all "products" for a given "region" via a key table. At this point we're essentially defining how products for a region will be retrieved later on using the key() function, but no work is actually being done. All that's accomplished here is that the XSLT processor is learning how to populate the key table so that later on in the process the key() function can be used to retrieve the elements.

The xsl:key element is defined as follows:

```
<xsl:key name="products" match=
 "product" use="region" />
```

There are three parts to the xsl:key element:

1. ***Name:*** The name that will be used to refer to this key table.
2. ***Match:*** The pattern that refers to the nodes that populate the key table. In this example "product" nodes are those that can be accessed using this key.
3. ***Use:*** This "use" expression defines the value associated with the node used in the key table. In this example the "product" nodes will be accessed using the "region" as the key. This will allow us to retrieve all of the "products" for a specific "region."

***Step 2:* Loop through the unique regions (the primary key) in the document.**

To fully appreciate what happens in Step 2, it's important to understand how the generate-id() function works: generate-id(nodeset) generates a unique identifier for the nodeset that's passed to it. It requires a nodeset as a parameter, but will return only the identifier for the first node (in document order) of the nodeset.

Note: It might appear on the surface that repeated calls to this function would produce a new generate-id() value each time they were called, thus causing the algorithm to fail. However, the same node will always return the same identifier during a single process. In a sense it behaves more like a "return-id" function than a "generate-id" function.

It's important to remember that values returned from generate-id() must be used blindly. There's no rhyme or reason

AUTHOR BIO

David Silverlight is chief XML evangelist for Infoteria Corporation (www.infoteria.com), an XML software development company based in Tokyo and Beverly, Massachusetts. He also maintains www.XMLPitstop.com, a resource for XML examples and everything else XML.

for the values generated other than that they be lexically valid as name tokens in XML. The processor is required to return the same ID for a given node every time it's requested in the running of a single process, but the processor isn't obliged to return the same value the next time around, even with the same data.

In some of the examples the values returned from the generate-id() function are displayed as part of the output. This is shown strictly to give a demonstration of what's happening under the covers. Please note that the generate-id() values should never be "interpreted" or used for analysis and/or comparison. You can't rely on the pattern generated since it will change from execution to execution, even if the data hasn't changed. The values are interesting to look at, however.

On to Step 2...

The end result of the xsl:for-each element below will be a return of product elements for the value of the region that's passed to it. More specifically, it's the result of examining all product elements and filtering out those whose "region" child values are duplicates, then filtering in the first element in document order with each unique value for a "region" child.

```
<xsl:for-each select="product
 [generate-id(.)=generate-id
 (key(‘products’,region))]">
```

The logic behind this statement isn't inherently obvious, but a further breakdown should clarify it. Essentially, the expression will return the elements from which the IDs from the product elements "generate-id(.)" match the IDs returned from the indexed key lookup function "generate-id(key('products',region))".

The goal in this equality test is uniqueness: find "only one" of each value, then act on the found ones. The technique accomplishes this requirement by using the key table to find "the first of each in document order" of each value. This works because (1) the key() function returns the set of nodes of the given value (in document order), and (2) the generate-id() function works on the first node (in document order) when given more than one node.

By combining (1) and (2) above, the result is the first node of each key table value in document order, thus giving us the first node representing each unique value.

The actual IDs generated from the two sets are given in Figure 1, which shows the IDs that correspond to both. Take a moment to examine this figure. It's important to observe that each call to generate-id(key('products',region)) will return a single value. However, when visually scanning the entire list of values displayed in Figure 1, it becomes apparent that there are only two unique IDs in the list (having values of "IDANHZDB" and "IDATGZDB"). On the other hand, the generate-id(.) expression will generate a unique ID for each product.

When visually scanning this column in Figure 1, we see that each element returns a unique value. When evaluating these two expressions in the equality test, only two elements are returned, producing only the unique "regions" from the XML document. Take a moment to look at the values in Listing 1 and Figure 1 to see what actual IDs are generated using these predicates. The downloadable source code for this article also contains this example and the code for Listing 2 (www.sys-con.com/xml).

***Step 3:* Display the products for each region.**

As each region is iterated through, the key() function comes into play once again. This time it's used as part of the "for-each" loop to iterate through the product elements for the given region. Rather than looping through the document, the key function uses the current region value to return a nodeset of all the corresponding product elements from the named key table. Though it's up to the processor to implement the key() function quickly, it's assumed the processor will obtain values from the key table much faster than revisiting the source node tree.

```
<xsl:for-each select
 ="key('products',
 region)">
```

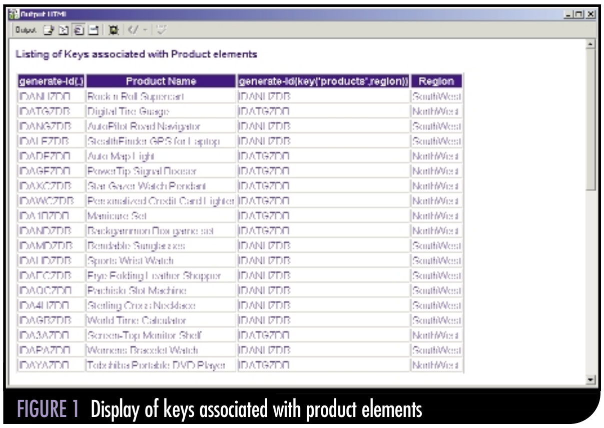
The expression "key('products'-,region)" will return all of the "product" elements from the key table whose "use=" expression defined in xsl:key (see Step 1) evaluated to the same value as the "region" child of the current element. In the example we specified use="region". If region has a value of "SouthWest", then all the product elements from the key table that contain a child element with a value of "SouthWest" will be returned.

Q: What are some of the cautions regarding the Muenchian Method?

A: The Muenchian Method requires an XML processor that supports keys, which not all parsers do. Although MSXML 3.0, Xalan, and Saxon support them, XT does not at the time of this writing. Alternative techniques that don't use the key() function do exist, but they're inherently slower based on the assumption that a processor will optimize the access to key tables (see **Elements of Design** section).

Keys can be memory intensive in their own way as a result of how they index document information. Using keys requires the information to remain in memory, thus demanding extra overhead for support.

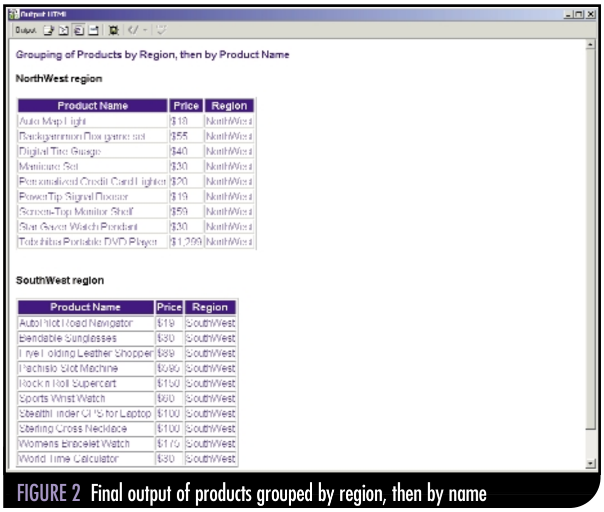
The complete code for the Muenchian Method example can be seen in Listing 2. Refer to Figure 2 to see the output generated.



Listing of Keys associated with Product elements

generate-id(.)	Product Name	generate-id(key('products',region))	Region
IDANHZDB	Rock n Roll Supercart	IDANHZDB	SouthWest
IDATGZDB	Digital Tire Guage	IDATGZDB	NorthWest
IDANGZDB	AutoPilot Road Navigator	IDANHZDB	SouthWest
IDALFZDB	StealthFinder GPS for Laptop	IDANHZDB	SouthWest
IDADFZDB	Auto Map Light	IDATGZDB	NorthWest
IDAGFZDB	PowerTip Signal Booster	IDATGZDB	NorthWest
IDAXCZDB	Star Gazer Watch Pendant	IDATGZDB	NorthWest
IDAWCZDB	Personalized Credit Card Lighter	IDATGZDB	NorthWest
IDA1DZDB	Manicure Set	IDATGZDB	NorthWest
IDANDZDB	Backgammon Box game set	IDATGZDB	NorthWest
IDAMDZDB	Bendable Sunglasses	IDANHZDB	SouthWest
IDALDZDB	Sports Wrist Watch	IDANHZDB	SouthWest
IDAFCZDB	Frye Folding Leather Shopper	IDANHZDB	SouthWest
IDAOCZDB	Pachislo Slot Machine	IDANHZDB	SouthWest
IDA4HZDB	Sterling Cross Necklace	IDANHZDB	SouthWest
IDAGBZDB	World Time Calculator	IDANHZDB	SouthWest
IDA3AZDB	Screen-Top Monitor Shelf	IDATGZDB	NorthWest
IDAPAZDB	Womens Bracelet Watch	IDANHZDB	SouthWest
IDAYAZDB	Toshiba Portable DVD Player	IDATGZDB	NorthWest

FIGURE 1 Display of keys associated with product elements



Grouping of Products by Region, then by Product Name

NorthWest region

Product Name	Price	Region
Auto Map Light	$18	NorthWest
Backgammon Box game set	$55	NorthWest
Digital Tire Guage	$40	NorthWest
Manicure Set	$30	NorthWest
Personalized Credit Card Lighter	$20	NorthWest
PowerTip Signal Booster	$19	NorthWest
Screen-Top Monitor Shelf	$59	NorthWest
Star Gazer Watch Pendant	$30	NorthWest
Toshiba Portable DVD Player	$1,299	NorthWest

SouthWest region

Product Name	Price	Region
AutoPilot Road Navigator	$19	SouthWest
Bendable Sunglasses	$30	SouthWest
Frye Folding Leather Shopper	$89	SouthWest
Pachislo Slot Machine	$696	SouthWest
Rock n Roll Supercart	$150	SouthWest
Sports Wrist Watch	$60	SouthWest
StealthFinder GPS for Laptop	$100	SouthWest
Sterling Cross Necklace	$100	SouthWest
Womens Bracelet Watch	$175	SouthWest
World Time Calculator	$30	SouthWest

FIGURE 2 Final output of products grouped by region, then by name

Elements of Design

A World Without Muench

This month's **Elements of Design** highlights other methods of grouping. The two methods showcased here were submitted by developers in the field. Each demonstrates a variation on the original message, has its own pros and cons, and simply offers alternative techniques for grouping XML data.

Grouping Without the Use of the key() function
(G. Ken Holman)

As mentioned in the preceding question, not all XSLT processors support the key() function. This solution becomes especially useful in scenarios in which the key() function isn't an option. Following are a few comments explaining the differences between Ken's method and the Muenchian Method, but to fully appreciate it, download the source code for the full example.

The essence of using the variable instead is very similar to a key table. In this solution all products are put into the variable (which itself is in document order):

```
<xsl:variable name="products" select=
 "//product"/>
```

Visit every product (which essentially happens when using a predicate à la Muench) in sorted order, but act only on the first product in document order (again using generate-id(), but this time with the variable instead of the lookup table):

```
<xsl:for-each select="$products">
  <xsl:sort select="name" order=
   "ascending"/>
 <xsl:variable name="region" select=
   "region"/>

  <xsl:if test="generate-id(.)=generate-
   id($products[region=$region])">
```

The rest is similar, except to grab the products of the same region from the variable instead of the key table:

```
<xsl:for-each select=
  "$products[region=$region]">
```

Grouping Using the Distinct Template
(EXSLT.ORG)

This solution varies from both Ken's approach and the Muenchian Method because it doesn't use key() or generate-id(). Instead, it uses the set:distinct template defined by the community initiative for commonly required extensions not available in XSLT.

This particular template can be used to return the distinct "region" elements instead of implementing the generate-id() function to return "product" elements with distinct "region" children. The focus changes slightly since we're obliged to deal with nodes being distinguished.

• • •

A working example of the solutions in this article can be downloaded from www.sys-con.com/xml.

DSILVERLIGHT@INFOTERIA.COM

LISTING 1 XSLT code used to generate output in Figure 1

```
<h2>Listing of Keys associated with Product elements </h2>
 <table border="1">
  <tr>
  <th>generate-id(.)</th>
  <th>Product Name</th>
  <th>generate-id(key('products',region))</th>
  <th>Region</th>

 </tr>
   <xsl:for-each select="product">
<tr>
  <td><xsl:value-of select="generate-id(.)"/></td>
  <td><xsl:value-of select="name"/></td>
  <td><xsl:value-of select="generate-
id(key('products',region))"/></td>
  <td><xsl:value-of select="region"/></td>
  </tr>
 </xsl:for-each>

 </table>
```

LISTING 2 Source code for the solution using the Muenchian Method of grouping

```
<?xml version='1.0' encoding='utf-8' ?>
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"  ver-
sion="1.0">
<xsl:output method="html"/>
<xsl:include href="Style1.xsl" />

<!--
Step 1: Define the primary key to be used in the Muenchian
grouping. The beautiful thing about the xsl:key  element in
our example is that once we know the "region," we can easi-
ly find all of the products that match that region.
The xsl:key element (different from the key() function) is
defined as follows:-->

<xsl:key name="products" match="product" use="region"/>

<!-- Template for our root rule -->
<xsl:template match="/">
 <xsl:apply-templates/>
</xsl:template>


<!-- Template for our "products" rule -->
<xsl:template match="products">

<xsl:call-template name="Style"/>

<h2>Grouping of Products by Region, then by Product Name
</h2>
<!--Step 2: Loop through the unique regions (the primary
key) in our document.  -->
<xsl:for-each select="//product[generate-id(.)=generate-
id(key('products',region))]">

 <!-- Sort Primary key by name in ascending order -->
     <xsl:sort select="name" order="ascending"/>

     <!-- Display the region as our table header -->
 <h3><xsl:value-of select="region"/> region</h3>

 <!--Display all nodes for a given region in a table-->
 <table border="1">
 <tr>
<th>Product Name</th>
  <th>Price</th>
  <th>Region</th>
 </tr>

<!-- For each value in our key collection for the given
region,
 display values -->

 <xsl:for-each select="key('products',region)">
<!--
The expression "key('products',region)" will return all of
the "product" elements from the key table whose "use="
expression defined in xsl:key (see xsl:key at top) evaluat-
ed to the same value as the "region" child of the current
element. In the example we specified use="region". If
region has a value of "SouthWest", then all of the product
elements from the key table that contain a child element
with a value of "SouthWest" will be returned.
-->
<!--Sort our secondary key, product nodes, by name-->
<xsl:sort select="name"/>
<tr>
  <td><xsl:value-of select="name"/></td>
   <td><xsl:value-of select="price"/></td>
   <td><xsl:value-of select="region"/></td>
  </tr>
     </xsl:for-each>
   </table>
 <br/><br/>
</xsl:for-each>

</xsl:template>

</xsl:stylesheet>
```

DOWNLOAD THE CODE @
www.sys-con.com/xml

We don't mind if they say we publish "geek" magazines, as long as we're known as...

The World's Leading *i*-Technology Publisher

June 2001
136,000 copies in print

June 2001 Preview
250,000 copies in print

June 2001
82,000 copies in print

June 2001
72,000 copies in print

June 2001
48,000 copies in print

Premiering December 2001
20,000 copies in print

Premiering January 2002
20,000 copies in print

Premiering February 2002
150,000 copies in print

and Producer of the World's Leading *i*-Technology Developer Conferences

JDJ EDGE conference & expo

XML EDGE conference & expo

web services EDGE conference & expo

COLDFUSION EDGE fasttrack

wireless EDGE conference & expo

Inc. 500 1999 WINNER

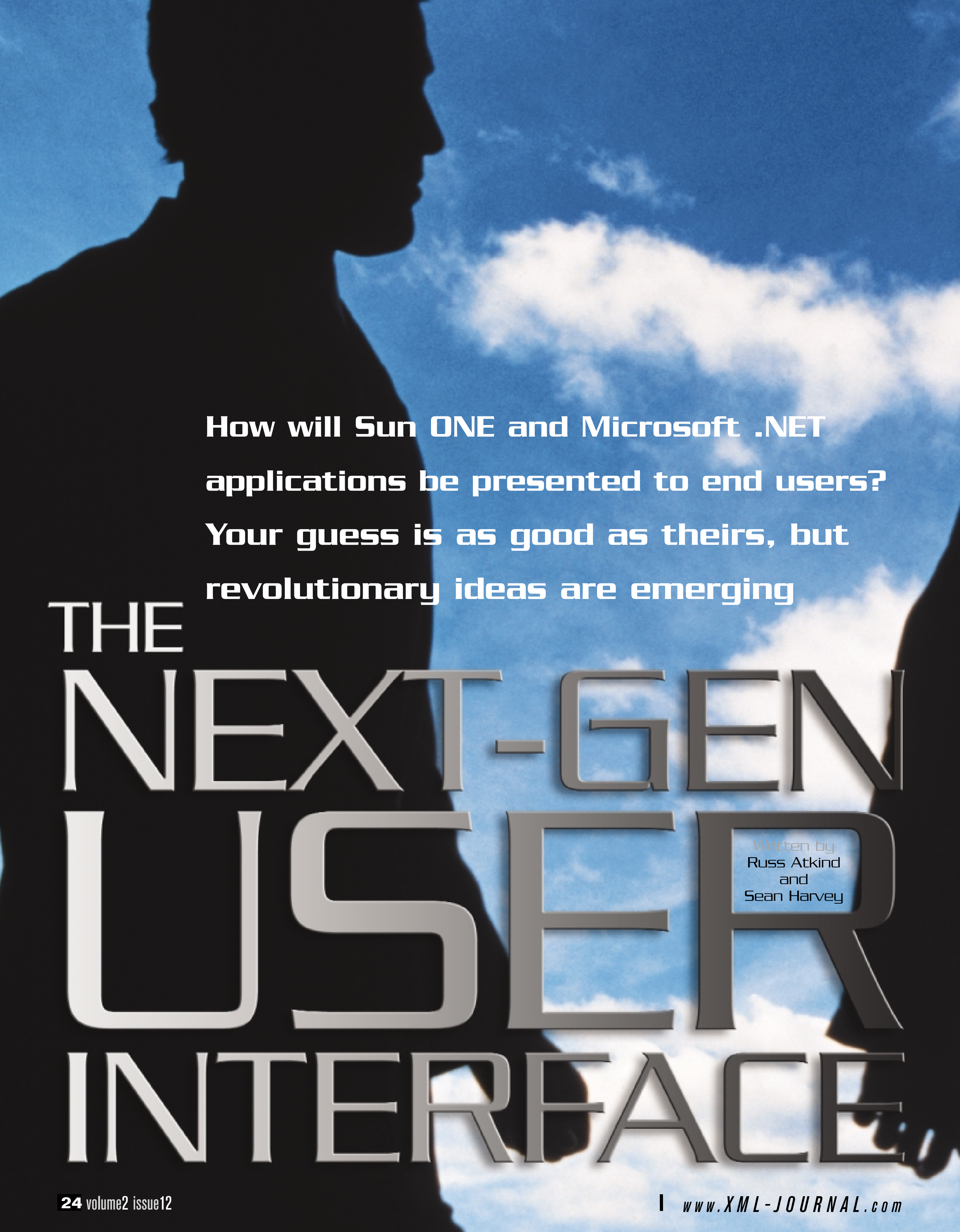
SYS-CON™ MEDIA

SYS-CON™ EVENTS

Inc. 500 2001 WINNER

www.SYS-CON.com

America's fastest-growing privately owned publishing company



THE NEXT-GEN USER INTERFACE

How will Sun ONE and Microsoft .NET applications be presented to end users? Your guess is as good as theirs, but revolutionary ideas are emerging

Written by
Russ Atkind
and
Sean Harvey

After months of hype, developers are finally sinking their teeth into the competing new initiatives by powerhouses Microsoft and Sun – MS .NET and Sun ONE. A beta version of .NET has been available on Microsoft's Web site since March, with Sun's beta platform due this spring.

At first glance the two projects are more notable for their similarities than their differences. Both seek the same two important goals: (1) universal back-end integration between every imaginable type of connected device via XML, and (2) moving desktop software onto the remote server, where it can be bought and sold on a per-use basis as a service.

Additional common goals include secure, universal access to your applications from any device and an IDE that allows you to create "smart" applications that react appropriately according to context (who you are, what platform you're on, etc.). And both support Java, to an extent. Even as Microsoft seeks to entice Java engineers away with C# and the Java User Migration Path (JUMP), Java applications can still take advantage of .NET Web services through XML and open standards–based protocols.

So far, so good. But what about the user interface? And what role will popular Java-based UI technologies like AWT, Swing, JSP, and servlets be assigned in the new order? Given both platforms' stated goal – the remote delivery of services that are every bit as good as the local, component-based software to which end users have grown accustomed – they're going to require a front end with a far richer user experience than is possible with today's standard client/server UIs. And they're probably not going to get it with the current HTML- and fat client–based paradigms.

In Figure 1 different kinds of applications are shown accessing Web services. Each application has a very different architecture and purpose. Components A and B are Web service providers. Component C is a desktop Java application that directly calls business logic in A and B. Component D is a multitier J2EE application. The business logic layer, implemented as EJBs, makes use of Web services. This fact is transparent to the presentation layer, implemented as servlets and JSPs. The UI is served to the end user in a browser.

RSATKIND@DROPLETS.COM

Russ Atkind is senior software developer at Droplet, Inc., where he ported the Droplets C++ API to Java, among other major development initiatives. Droplets is a thin-client, browser-independent UI delivery platform for serving remote applications with the functionality and speed of desktop software. Russ holds a BA in computer science from Yale University.

SHARVEY@DROPLETS.COM

Sean Harvey, senior technical writer at Droplet, Inc., developed the Droplets SDK platform documentation. He previously worked for Tempest Software, Inc., and the UN. He is the author of several travel books for the Rough Guides series and has been published in numerous journals covering a variety of fields.

> "Both Sun and Microsoft are reacting to the limitations of current UI paradigms by making their initiatives as UI agnostic as possible"

Defining the UI of the Future

Client/server UI development has been extremely Web-centric in recent years, enabling an entire generation of thin-client, page-based GUIs that are served through technologies such as Java servlets. All business logic remains on the server, allowing for centralized administration and automatic upgrades, and giving end users secure access to their applications from any computer with a Web browser and an active Internet connection.

But these applications have serious problems related to the page-based nature of the Web. HTML was originally designed as a medium for the transmission of text, and while servlets enable these pages to be served as business logic dictates, the UI itself is less attractive and functional than local software.

Imagine, for example, trying to use Napster if it were an HTML-based servlet application. There would be none of the asynchronous, dynamic UI updates that allow you to follow the progress of each download as it proceeds. You'd have to wade through page after page in order to perform even the simplest tasks, response time would be much slower, and high-level functionality like Chat would be impossible. Napster's GUI may be a bit of an ugly duckling, but its presentation as an asynchronous piece of local software gives it a number of functional advantages over any HTML-based competitor.

But fat client GUIs like Napster have their own set of problems. Unlike Web-based solutions such as Hotmail, they don't allow you instant access to your account from any computer anywhere in the world, and administration is difficult. Upgrades must be done manually on every machine, forcing developers to account in the business logic for every legacy version that may still be in use.

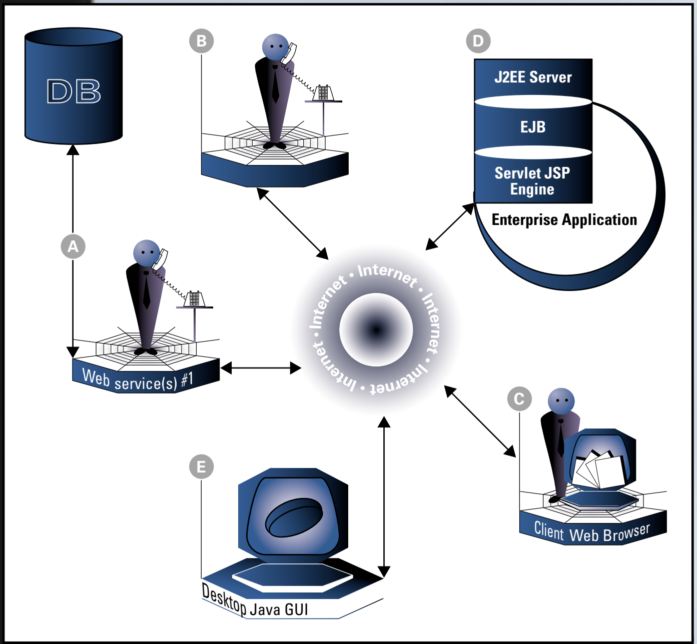


FIGURE 1 Different kinds of applications accessing Web services

What should the UI of the future look like? If Microsoft and Sun are to deliver on their promises, it would have to combine the benefits of Web-based and fat-client applications while simultaneously avoiding their many problems. As with servlet-based applications, the new UI will have to be:

- Universally accessible
- Easy to administer
- Secure

And like fat-client software, the new UI will have to be:

- Instantaneously responsive
- Easy to use
- Asynchronous
- Persistent
- Component-based

UI Agnosticism

Both Sun and Microsoft are reacting to the limitations of current UI paradigms by making their initiatives as UI agnostic as possible. Rather than lock themselves into a particular experimental scheme, they've chosen to ensure that their initiatives integrate with the existing Web-based and fat-client alternatives, despite their clear limitations – and will wait to see how the remote GUI evolves before placing any bets.

This means that – in the first incarnation of both Sun ONE and .NET – Java developers will have the freedom to work with the existing UI paradigm of their choice. Sun ONE will speak Java as its language of choice, while Java presentation logic will be able to interoperate with .NET-based business logic, provided it speaks XML and the open standards–based protocols that Microsoft uses on top of it.

While Microsoft's JUMP initiative has been grabbing all the headlines, this is simply a way to give Java code access to Microsoft's answer to the JVM, the Common Language Runtime (CLR). But running on CLR is not at all necessary for Java to interoperate with .NET. All you need do is enable your Java front end to speak with the .NET back end using its open standard–based protocols on top of XML (see Table 1).

As far as new UI initiatives, Microsoft has contented itself with an upgrade of its Active Server Pages, now called ASP.NET – though use of this tool isn't required for presenting .NET applications. The new version of this familiar technology offers a dramatically different program model from both its predecessor and JSP, bringing Web programming a step closer to GUI toolkit programming. It boasts a new Web Forms class library that gives you programmatic access to the elements on a page; buttons and tables are exposed as objects whose properties can be manipulated. In addition, rather than force the developer to interpret the raw name/value pairs submitted as a result of an HTTP POST or GET, ASP.NET translates the user's actions into events delivered straight to the components' event listeners. Combine this with ASP.NET's automatic preservation of the state of each page from one HTTP request to the next, and the development paradigm looks very much like that of AWT and Swing. But the inherent limitations of Web-based applications still prevent ASP.NET from competing with the responsiveness and asynchronous dynamism of local software.

> “As far as new UI initiatives, Microsoft has contented itself with an upgrade of its Active Server Pages,now called ASP.NET”

Swung?

Sun ONE’s first release also strives to be as UI agnostic as possible, and will allow you to use all of the familiar Java-based UI frameworks. Consider, for example, Sun ONE’s Webtop, a framework for developing productivity software that can be served remotely. Webtop is concerned almost exclusively with providing back-end services that can be called from the UI of your choice. This is good news for Java developers well versed in the current technology. Those with servlet and JSP experience can still leverage their skills in a new generation of Web-based software, while those wishing to write fat-client front ends will still be able to leverage Swing and AWT.

But in recognition of the limitations of these existing interfaces, Sun is also strongly considering the integration of a completely new (and very different) UI paradigm into Webtop – XUL (pronounced *zool*), an open-source, tag-based language created by Mozilla, and used to render Netscape 6. If their effort turns XUL into the next-generation UI framework of choice, developers may find their familiar Java GUI tools quickly marginalized.

XUL is not quite like anything else UI developers have worked with before, but its component technologies will seem familiar. Like HTML, the widgets in a window are described using XML. Cascading Style Sheets (CSS) provide the appearance parameters, and JavaScript is used to handle events and invoke business components. But the end result is quite different: XUL renders component-based GUI applications that behave like traditional desktop software.

Given that Sun has pledged to make Java the main application-building technology for Sun ONE, XUL might seem a surprising choice. In its original form XUL uses JavaScripts to call business logic in C++ components. Fortunately, these components adhere to a cross-platform binary standard (XPCOM), and a Sun-supported project is underway to achieve full interoperability with Java.

Remote XUL?

Is XUL the UI of the future? If it could be served remotely via a thin client (like HTML), it could solve a number of “Web services” UI delivery problems. Unfortunately, its bandwidth requirements are extremely high and it can’t provide the asynchronous UI updates to support interactive applications like stock watchers or IM. But even before these key limitations are addressed, the issues involved with keeping the XUL application’s business logic on the server will have to be ironed out.

There are several obstacles. First, to be a truly ubiquitous remote UI platform, a UI needs to adapt for display on different devices. While XUL shows promise in this area because of the inherent transformability of XML, it currently requires an extremely heavyweight client-side engine. In addition to an HTML renderer, the engine must embed a JavaScript interpreter and an XML parser, and support for CSS, the XUL language, and the XPCOM component model – essentially the entire Mozilla browser.

Second, there needs to be a way to communicate between the front and the back end. Unlike HTML forms, XUL doesn’t automatically send the data entered by the user back to a server whenever the user clicks on a widget. This is in fact an advan-

Protocol	Description	Homepage/Specification	Java Support
Universal Description, Discovery and Integration (**UDDI**)	Used to look up and locate Web services	www.uddi.org	uddi4j on IBM’s developerWorks; Bowstreet’s open source implementation: www.juddi.org; JSR-000093 (Java API for XML Registries)
Web Service Description Language (**WSDL**)	Used to describe the interface of a Web service	www.w3.org/TR/wsdl	IBM on alphaWorks; proposed for the Java platform as JSR-000110
Simple Object Access Protocol (**SOAP**)	Used to invoke methods on Web services	www.w3.org/TR/SOAP	http://xml.apache.org/soap/; JSR-000101 (Java APIs for XML RPC)
Disco	Microsoft’s own protocol for discovering the services on a given server	http://msdn.microsoft.com/xml/c-frame.htm?/xml/general/disco.asp	None. Microsoft alone developed Disco. Unclear how this will affect interoperability.

TABLE 1 Tools for integrating Java with .NET protocols

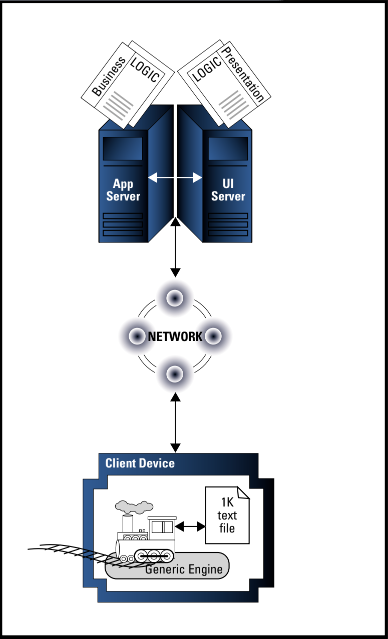


FIGURE 2 UEP application delivery architecture

> “If Sun's effort turns XUL into the next-generation UI of choice, developers may find their familiar Java tools quickly marginalized”

tage, since there’s potentially much finer-grained control over how and when that data is sent. But the plumbing has yet to be built.

One potential approach would be to make XPCOM components remotely callable. In its present form, the XPCOM specification, while not preventing remote calls from taking place, doesn’t provide a framework to support it. The developer would have to take care of all marshaling and networking issues – an unacceptable burden in today’s world. If remote proxies for components could be automatically generated, the JavaScript that accompanies the XUL code could transparently invoke business logic on the server.

Work is also in progress to allow control code itself to be written in XPCOM components, and therefore in a language other than JavaScript. If mozilla.org provides an XPCOM-compliant interface for event handlers, event-handling code itself could potentially be a remote component invoked directly when the user performs an action, such as the click of a button. This would enhance security by keeping the client from downloading executable code. In addition, the client device would no longer need a JavaScript interpreter, significantly decreasing client-side footprint and processor consumption.

Streaming Applications

But wouldn’t it be nice to keep writing AWT and Swing-like Java front ends, just as you always have? While Microsoft and Sun are focusing on the browser, a group of start-ups is attempting to find new ways to deliver Java GUIs to the desktop, bypassing both the sluggishness of Web applications and the long download times of large applets and fat clients. One such method, the “streaming applet,” delivers application code to the client in a mode analogous to a media player. Developers write AWT or Swing front ends more or less as if they were served locally. The application server then automatically chops the application into chunks and serves them to the client as needed. The server attempts to make intelligent decisions about which pieces to deliver first, based on typical usage patterns. Once the Java logic arrives, it’s cached (but not installed) and run on the client side, giving end users a centrally administered, locally processed application. Developers needn’t worry about network issues.

Of the two companies we’re aware of that are pursuing this strategy, AppStream and Softricity, neither, to date, has delivered a beta version of their product to the public. Such a paradigm would have to address a number of inherent difficulties. For one thing, delivering an entire application over the network is still bandwidth-intensive, whether whole or bit by bit. And a severed network connection might be difficult to handle since it could leave some part of the application partially downloaded. Finally, the logic to decide when to deliver each piece of the application would have to be quite intelligent. Since the technique is billed as a “no additional programming required” solution, little specific information about each application is likely to be available to the framework.

Remote Delivery, Local Appearance

Our company, Droplet, Inc., has developed a less bandwidth-intensive way to accomplish a similar goal. Droplets serve an AWT-style UI remotely without transferring any of the application code to the client. This allows developers to write and serve thin-client applications using Java’s highly evolved, well-designed GUI development paradigms. Developers write the UI almost exactly as if they were building local software, but the application remains on the server and can be used from anywhere on the Internet.

Since all application logic resides on the server, the only necessary client-side piece is a generic, lightweight client. The user isn’t aware of the client; all served UIs looks exactly like locally installed applications. A two-way, asynchronous protocol – we call it a User Event Protocol (UEP) – transmits UI update instructions from server to client and user actions from client to server (see Figure 2). Whereas a Web browser must receive an entire page in response to a user action, a Droplet receives only the changes necessary to update its display.

Limiting client/server communication to events and updates results in a lightweight protocol that increases application responsiveness dramatically and performs well over low-bandwidth devices. Since the client and server maintain a two-way connection, UEP can also deliver asynchronous UI updates to the client, enabling applications like Napster and instant messaging. This capability is crucial to many applications but eludes Web-based platforms. The protocol can also tunnel through HTTP when necessary, still supporting asynchronicity through efficient client polling. The fact that HTTP tunneling is sometimes necessary is thus hidden from the application programmer.

The advantage of this framework to Java developers is the ability to write remotely served, locally presented applications to end-user desktops using the AWT/Swing paradigm they’re already familiar with. It also solves a number of engineering problems that occur in Web development: it hides all networking code, eliminates the need to track per-user session state (values of instance variables don’t disappear until the user exits the application), and eliminates Web programming problems caused by the browser’s back button – all while retaining acceptable speed and full interactivity.

Conclusion

The advent of Web services – and their whole-hearted adoption by the world’s two most important software companies – requires a far more advanced UI framework than has been developed to date, and it’s too early to tell which of the new competing paradigms will win out. While Sun’s adoption of the tag-based XUL language and Microsoft’s enhancement of Active Server Pages have cast a small cloud over the future of Java UI development, a number of companies have come up with innovative ways to leverage the sophistication of AWT and Swing as front ends for Web services. Meanwhile, the software giants are hedging their bets, lest their initiatives be outflanked by the rapid transformations now taking place in remote UI delivery.

Resources

- ***Appstream home page:*** www.appstream.com/
- ***ASP.NET home page:*** www.asp.net/
- ***Droplets Remote GUI Toolkit for Java:*** www.droplets.com/scripts/sdk.asp
- ***Project Blackwood: Integration of the Java Platform with XUL:*** www.mozilla.org/projects/blackwood/
- ***Softricity home page:*** www.softricity.com/
- ***Sun ONE Webtop home page:*** www.sun.com/sunone/

FOR WLS DEVELOPERS BY WLS DEVELOPERS
BEA
WebLogic
DEVELOPER'S JOURNAL
An introduction to...

e-BUSINESS SOLUTIONS: TAKING BUSINESS TO THE NET
SPECIAL: PREMIER ISSUE!
WebSphere
DEVELOPER'S JOURNAL
INTRODUCTORY CHARTER SUBSCRIPTION
SAVE $30 Today!
Iterative Development with VisualAge for Java
Best Practices in WebSphere Application Packaging and Deployment
IBM WebSphere Studio v3.5 Advanced
JDBC Connections

[WRITTEN BY BOB MCWHIRTER

XPath evaluation against any object model in Java

Introduction to SAXPath and Jaxen

The W3C (http://w3c.org/) defined XML as a data model. Soon thereafter, work was started to define XPath, the language for addressing parts of an XML document. XPath isn't a technology for performing queries upon an XML document – that's the realm of the XQuery specification – but rather a simpler method for addressing or matching parts of a document.

In the world of XML document usage there's a clear separation between the parser, which can recognize XML tokens and validate to some extent the semantics of the document (using a DTD or a schema), and the application, which uses the data. Interfaces such as the W3C DOM provide an object tree–based representation of the document to the application programmer, whereas the SAX (the Simple API for XML) provides a callback method for interacting with a document in a top-to-bottom event-stream method.

With SAX, events are fired at the start of each <tag>, for any character data or nested tags contained therein, and at the end of each <tag>. Additionally, events are signaled for other entities, such as processing instructions and comments.

While XML parsing technology has come a long way, providing multiple methods for interacting with a document, the same can't be said for other, related technologies, like XPath. Typically, the parsing of an XPath expression is closely linked to the application using it, typically an XSLT processor. But as XPath becomes embedded in more and more XML specifications, such as C14N (XML Canonicalization) and XPointer, having a clean separation between parsing and using XPath becomes more valuable. This is the aim of the SAXPath project. Additionally, with the proliferation of different specialty object models for representing XML documents, each one currently requires its own implementations of an XPath and XSLT engine. The aim of the Jaxen project is to act as a buffer between XPath expressions and object models, allowing a single XPath implementation to operate on many models. Jaxen is one of the first projects to be built on the SAXPath API.

SAXPath

SAXPath is modeled closely on the structure used by David Megginson with SAX. The two most commonly used interfaces are org.saxpath.XPathReader and org.saxpath.XPathHandler.

Applications that wish to handle the parse events must implement the XPathHandler interface, which receives events from a parser that implements the XPathReader interface. More generally, the XPathReader interface extends the SAXPathEventSource, which allows SAXPath events to come from basically any source and not directly from a string parser. Any parser that correctly implements the XPathReader interface should be able to plug and play with your application seamlessly.

By default, the com.werken.saxpath.XPathReader parser is included with the SAXPath distribution, though any other available ones may be used.

Getting Started

To use SAXPath, you must first be able to instantiate a parser. This is done with the aid of a helper class, which instantiates a parser either directly, through a class name parameter, or by using a Java property or the default implementation.

First you must import the necessary helper class, the XPathReader interface, and the exception class:

```
import org.saxpath.XPathReader;
import org.saxpath.SAXPathException;
import
org.saxpath.helpers.XPathReaderFactory;
```

Next, use one of the createReader() methods. To instantiate, use either the default implementation or, based on the class name provided, the org.saxpath.driver system property – whichever is the simplest form (see Listing 1).

This method examines the org.saxpath.driver property for a fully qualified class name (such as com.werken.saxpath.XPathReader). If that property isn't set, the default parser class name is used.

If you wish to have direct control from your application, simply use the flavor of createReader() that takes a String class name parameter:

```
XPathReader reader =
XPathReaderFactory.createReader
("com.werken.saxpath.XPathReader");
```

Before you can do anything useful with the new XPathReader, you must register an XPathHandler implementation with it.

```
XPathHandler handler =
new MyXPathHandler();
reader.setXPathHandler( h2andler );
```

All that's required now to receive parse events from the XPathReader is to pass in an XPath expression for parsing, potentially catching required exceptions (see Listing 2).

Your XPathHandler will now receive events matching a recursive-descent parse of the XPath expression.

For more information about working directly with the SAXPath parse events, go to http://saxpath.org/.

Jaxen

Motivation

I initially created werken.xpath to be an XPath engine to support Jason Hunter and Brett McLaughlin's JDOM project. When James Strachan created dom4j, he also ported the werken.xpath library to his new object model. This was

AUTHOR BIO

Bob McWhirter is an open-source developer who has created and contributed to several open-source projects, including ANTLR, jakarta-velocity, werken.opt, werken.xpath, SAXPath, and Jaxen. He's a member of the JDOM JSR Expert Group.

SHOP ONLINE AT **JDJSTORE.COM** FOR BEST PRICES OR CALL YOUR ORDER IN AT **1-888-303-JAVA**

JDJ STORE.COM GUARANTEED LOWEST PRICES!

Guaranteed Best Prices

JDJ Store Guarantees the Best Prices. If you see any of our products listed anywhere at a lower price, we'll match that price and still bring you the same quality service.

Terms of offer:

- Offer good through December 31, 2001
- Only applicable to pricing on current versions of software
- Offer does not apply toward errors in competitors' printed prices
- Subject to same terms and conditions

Prices subject to change.
Not responsible for typographical errors.

Attention Software Vendors:

To include your product in JDJStore.com, please contact tony@sys-con.com

SYS-CON MEDIA

XML-Journal: The Complete Library

JDJStore.com $53⁹⁹

SYS-CON MEDIA

JDJ, CFDJ, XML-J, CFAdvisor The Complete Works

Collect All Four!

JDJStore.com $229⁹⁹

SYS-CON MEDIA

CFDJ: The Complete Library

JDJStore.com $71⁹⁹

MACROMEDIA®

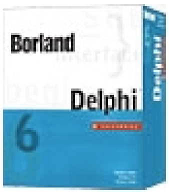
HomeSite™ 4.5

HomeSite 4.5™ from Macromedia (HomeSite) is the award-winning HTML editing tool that lets you build great Web sites in less time, while maintaining pure HTML. Created by Web developers for Web developers, only HomeSite gives you precise layout control and total design flexibility, while delivering the latest Web technologies.

HomeSite 4.5 . . . . $93⁹⁹

BORLAND

Borland Delphi 6 Enterprise



Delphi 6 makes next-generation e-business development with Web Services a snap. BizSnap Web Services development platform simplifies business-to-business integration by easily creating Web Services. DataSnap Web Service-enabled middleware data access solutions integrate with any business application. Rapidly respond to e-business Web presence opportunities ahead of the competition with WebSnap, Delphi's complete Web application development platform.

Borland Delphi 6 Enterprise $2849⁹⁹

SYBASE

Adaptive Server Enterprise for WINNT v12.0

Sybase Adaptive Server Enterprise 12.0 is designed to support the demanding requirements of Internet and traditional, mission-critical OLTP and DSS applications. The efficient multithreaded architecture, internal parallelism and efficient query optimization of Adaptive Server Enterprise delivers unsurpassed levels of performance and scalability.

Adaptive Server Enterprise WINNT v12.0 . . . . $798⁹⁹

POLYCOM

VoiceStation Personal Teleconferencer

Designed to improve your business communications while increasing productivity, VoiceStation 100 enables you to conduct remote meetings that are as natural as speaking face-to-face. Features: full-duplex audio performance featuring Polycom's Acoustic Clarity Technology™, dynamically adjusts to any small room's acoustic environment, 3 microphones w/ 360° room coverage, high-quality speaker, easy to install / easy to use, redial, flash, hold & mute, data port (110V versions only), RCA output jack.

VoiceStation Personal Teleconferencer $279⁹⁹

CASIO

EM500 Multimedia Cassiopeia

The new slim-designed EM-500 is a mobile multimedia tool that targets a younger market. Designed to highlight Casio's new faster processor, and available in five different colors, the EM-500 is stylishly designed and engineered to take advantage of the new and emerging digital content that is available on the Internet. (For online browsing and e-mail, an optional modem is required and sold separately.)

EM500 Multimedia Cassiopeia $428⁹⁹

eHELP

RoboHELP Office 2000

RoboHELP Office provides a user-friendly WYSIWYG authoring environment for creating JavaHelp. RoboHELP guides you through the process so you can create a great JavaHelp system with point-and-click and drag-and-drop ease. Now you can create JavaHelp systems as easily as you create WinHelp, Microsoft HTML Help and WebHelp (cross-platform Help) from the same source product – all with RoboHELP Office.

RoboHELP Office 2000 . . . . $898⁹⁹

ORDER TODAY! 888-303-JAVA

effectively a fork of the original werken.xpath code, and created maintenance headaches as it became difficult to migrate bug fixes between the dom4j and JDOM versions.

Upon further inspection, James and I realized that XPaths are specified only in terms of the XML InfoSet for retrieving and navigating an XML document. With this realization, Jaxen was created as an XPath engine that works with object model adapters that provide a uniform InfoSet-centric view of any object model. Through this, a single core XPath engine could be maintained for many models, with only a thin adapter required.

Getting Started

While Jaxen uses SAXPath under the covers to parse XPath expressions, this fact remains mostly hidden to the user. Each model supported by Jaxen has its own package (such as org.jaxen.dom4j.* or org.jaxen.jdom.*). Once you determine the correct package to use for your model, all other code is identical. To use a different model, simply adjust the import statements as needed. To use Jaxen with dom4j documents, the required import statement is:

```
import org.jaxen.dom4j.XPath;
```

For JDOM, you'd use:

```
import org.jaxen.jdom.XPath;
```

In either case, all other code is the same (see Listing 3).

LISTING 1

```
public static void main(String[] args)
{
     try
     {
          XPathReader reader =
XPathReaderFactory.createReader();
          // ... code that uses the XPathReader
here ...
     }
     catch (SAXPathException e)
     {
          e.printStackTrace();
     }
}
```

LISTING 2

```
try
{
     reader.parse(
"/article//section[@name='getting-started']");
}
catch (XPathSyntaxException e)
{
     System.err.println( e.getMultilineMessage()
);
}
catch (SAXPathException e)
{
     System.err.println( e.getMessage() );
}
```

LISTING 3

```
try
{
     XPath xpath = new XPath("/article//sec-
tion[@name='getting-started']");
     // ... code using the xpath goes here ...
}
catch (XPathSyntaxException e)
{
     System.err.println( e.getMultilineMessage()
);
}
catch (SAXPathException e)
{
     e.printStackTrace();
}
```

DOWNLOAD THE CODE @
www.sys-con.com/xml

XPath objects are fully reentrant and thread-safe. They contain no internal state for evaluation and thus can be cached easily and shared within an application. Once you have an XPath object, you can apply it against various initial contexts and retrieve results in several different ways:

- You can select a single node (which selects only the first matching node for the given expression):

```
xpath.selectSingleNode
( initialContextObject );
```

- You can select all matching nodes:

```
xpath.selectNodes
( initialContextObject );
```

- You can select a Number interpretation of the expression:

```
xpath.numberValueOf
( initialContextObject );
```

- You can select a simple String value interpretation of the expression:

```
xpath.valueOf( initialContextObject );
```

Abstracting Away the Object Model

Jaxen was deliberately designed to be flexible, open, and useful for many purposes. Through the use of the Adapter pattern, implemented through the Navigator interface, virtually any object model can be accommodated. The interface that Jaxen needs from an object model is basically that of the W3C-InfoSet specification. Thus Navigator has methods corresponding to many aspects of the InfoSet.

Through the use of Java's own Iterator pattern, which provides access to the various XPath axes, Jaxen causes little impact on the performance of each model. It would be awkward and inefficient to require collections to be expressed as a specific type, such as java.util.List, because some models, such as DOM, have their own collection objects, such as NodeList. By requiring only the much simpler contract provided by a read-only Iterator, Jaxen doesn't introduce additional inefficiencies.

The Navigator mechanism, along with the already developed implementations, could also be useful in other applications. Apache-Xalan, an XSLT processor, for example, currently supports DOM trees handily. Other types may be accommodated by using implementations of javax.xml.transform.sax.SAXSource, but not natively. It should be possible to rework Xalan to use the Navigator interface and support many models natively.

As already mentioned, using the SAXPath event API maintains loose coupling between the parsing and evaluation components, which increases reusability.

Future Directions

James Strachan is currently working on a new project, betwixt, which will be able to provide an XML representation of arbitrary JavaBeans. Additionally, he'll be providing an implementation of Navigator that will use betwixt to allow XPath expression evaluation on JavaBeans. Interest has been expressed for a Jaxen-based XQL engine.

Jaxen can certainly be integrated into many existing applications that need only lightweight XPath evaluation and not an entire XSLT engine. For example, David Megginson's NewsML Toolkit uses Jaxen on DOM trees.

Since both SAXPath and Jaxen use a flavor of the BSD license, developers are free to use them in both open-source and commercial projects without limitation.

Getting Involved

Both SAXPath and Jaxen are open to contributors, and are hosted at SourceForge (http://sourceforge.net/). Jaxen, particularly, needs users of different, currently unsupported, object models to create the DocumentNavigator buffer to allow for full XPath support. The SAXPath API could easily be retargeted to other languages, such as C++ or Python.

Resources

- ***SAXPath Web site:*** http://saxpath.org/
- ***SAXPath SourceForge project:*** http://sourceforge.net/projects/saxpath
- ***Jaxen Web site:*** http://jaxen.org/
- ***Jaxen SourceForge project:*** http://sourceforge.net/projects/jaxen/
- **dom4j *Web site:*** http://dom4j.org/
- ***JDOM Web site:*** http://jdom.org/
- ***EXML Web site:*** http://themindelectric.com/
- ***W3C XPath Specification:*** www.w3.org/TR/xpath

BOB @ WERKEN.COM

2002

JUNE 24-27

JACOB JAVITS CONVENTION CENTER

NEW YORK, NY

JAVA, XML AND .NET TECHNOLOGIES

James Gosling,
The Father of Java

web services EDGE conference&expo™

INTERNATIONAL WEB SERVICES CONFERENCE & EXPO

JDJ EDGE conference&expo™

INTERNATIONAL JAVA DEVELOPER CONFERENCE & EXPO

XML EDGE conference&expo™

INTERNATIONAL XML CONFERENCE & EXPO

Fundamentally Improving the Speed,
Cost & Flexibility of Business Applications

Who Should Exhibit...

Java, XML, Web services and .NET Technology vendors, staking their claim to this fast-evolving marketplace.

JDJEdge 2002 and Web Services Edge East 2002 Will Feature...

- Unmatched Keynotes and Faculty
- Over 150 Intensive Sessions and Fast Tracks
- The Largest Independent Web Services, Java and XML Expos
- An Unparalleled Opportunity to Network with over 5,000 *i*-technology professionals

Who Should Attend...

- Developers, Programmers, Engineers
- i-Technology Professionals
- Senior Business Management
- Senior IT/IS Management
- Analysts, Consultants

THE CALL FOR PAPERS IS NOW OPEN
VISIT WWW.SYS-CON.COM FOR DETAILS

FOR MORE INFORMATION

SYS-CON EVENTS, INC.
135 CHESTNUT RIDGE ROAD
MONTVALE, NJ 07645

201-802-3069
WWW.SYS-CON.COM

SYS-CON MEDIA
OWNED & PRODUCED BY
SYS-CON EVENTS

SHOW REPORT

[WRITTEN BY **JEREMY GEELAN & SEAN RHODY**

web services EDGE conference&expo

XML EDGE conference&expo

ConferenceConsensus

WEB SERVICES – TAKING OVER THE *IT* WORLD

Web Services Edge West and XML-Edge 2001 opened for business October 22 with a series of sessions and tutorials. Five tracks were maintained throughout the conference.

The conference proper began on Tuesday, October 23, with a keynote presentation by Dr. Charles Goldfarb, the father of XML technology. Goldfarb's presentation explored the dichotomy between application developers and document developers as they approach the concept of XML from completely different perspectives. XML "is a precision tool...being misused by people who think it's a shotgun," he asserted.

He expressed this dichotomy with a number of examples that led to the development of a concept he calls HARP – Human, Abstraction, Rendering, and Presentation – and made a strong case for extracting and encapsulating data in XML format.

To reinforce his point about how revolutionary XML is, Goldfarb took the audience through the history of data systems. But delegates' interest level increased markedly as he asserted that many so-called XML "gurus" in fact misuse or manipulate some of the basic terminology associated with XML in order to sell XML-based products of one variety or another.

Goldfarb's vision for XML is pure and based on freedom, not proprietary solutions. "It frees developers from having to be locked into a single processing paradigm," he explained.

"XML is based on free and open standards. So every time I see that phrase *MSXML*, I get a little nervous," he quipped, in a reference to the initials of a parser that bear a wholly coincidental similarity to those of a certain company in Seattle with a well-known reputation for keeping things proprietary.

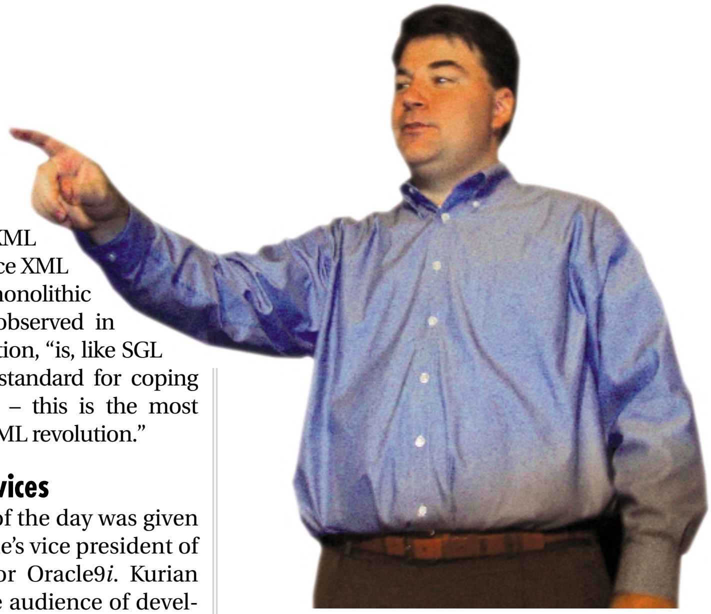
Goldfarb also told the audience at his keynote that he expected to see large companies like SAP adopting XML wholesale "very soon" since XML removes the need for monolithic data systems. XML, he observed in rounding off his presentation, "is, like SGL before it, designed as a standard for coping with the unstandardized – this is the most important aspect of the XML revolution."

Java-Driven Web Services

The second keynote of the day was given by Thomas Kurian, Oracle's vice president of product development for Oracle9*i*. Kurian explained to an attentive audience of developers and *i*-technology professionals how greatly they would benefit "if the J2EE and Web services programming models were rendered analogous so that you didn't have to learn a new programming model." Kurian outlined Oracle's vision for a world of Java-driven Web services and presented in detail how the architecture of, for example, Oracle's own 9*i*AS application server with its unified model lends itself to building, deploying, and managing Web services.

"There's a lot of hype going round," he observed, "about Web services as nirvana." Oracle's view, according to Kurian, was more grounded, and based on a solid belief that open standards and interoperability can take e-business to the "next level," as he termed it. By that he meant the next level of operational efficiency, and – therefore – of profitability.

He explained that in his view UDDI, SOAP, and



Steve Benfield CTO, SilverStream Software

Santa Clara Convention Center

CEO panel participants. Front *(l to r)*: Ali Kutay, former CEO, WebLogic, President/CEO, AltoWeb; Eileen Richardson, President/CEO, Infravio; Barry Morris, CEO, IONA Technologies. Rear *(l to r)*: Annrai O'Toole, Chairman, Cape Clear; Sean Rhody, panel moderator, Editor-in-Chief, *Web Services Journal*; Dirk Slama, CEO, Shinka Technologies; Greg O'Connor, President, Sonic Software.

Steve Benfield, CTO, SilverStream Software

WSDL alone will not be enough, and how the true success of more complex Web services will require ebXML, RosettaNet, and the Java JAX standards.

"Web services is a great way to allow applications to be built. It's a set of technologies to let one app talk to another using open standards," Kurian continued. "Oracle believes in open standards and interoperability, including .NET, which is why our goal is to unify the J2EE and Web services models."

On Wednesday, October 24, a large show crowd explored the wares of more than 40 vendors, including SilverStream, Borland, and IONA. Several vendors unveiled new products, including Systinet (formerly IDOOX) and Borland.

The morning's opening keynoter, Steve Benfield, CTO of SilverStream Software, addressed the overall question of Web services and emerging application developer techniques.

Well known in the industry for his view that "Web services will change the world," Benfield nonetheless took care to explain that in many ways "there is nothing magic about Web services." Their impending success is based on changes in the overall business environment as much as on any quantum leap in technology. Benfield asked, "What is all the fuss about?"

The answer, he asserted, is that while developers have been doing XML over HTTP for a while, Web services comprises far more than that. A Web service, he said, can best be defined as "a readily shareable business function," which is what makes the move toward Web services revolutionary and constitutes a business opportunity for software vendors and business alike since it will help customers extend their business to the Web.

Flexibility is another advantage. "A few years ago, when you built an application, you knew who the users were going to be and what they'd use, but today you don't. Web services creation allows you to architect for reuse, which can be done without regard to the actual user."

The most important difference between Web services and previous attempts – like CORBA – to ensure interoperability between businesses is that every major vendor is working to support it: "With Microsoft and IBM both behind it 100%," Benfield said, "Web services is uniquely positioned to succeed. Plus, it works using existing Internet and XML standards."

Benfield explained that "mere mortals" won't be able to build successful Web services without help, and that a new product set will sweep through the industry, what he termed *integrated services environments* – in other words, an entire framework for building, deploying, and managing Web services.

"If you want to spend your time building applications versus understanding the semantics of Web Services Description Language and Simple Open Access Protocol, then you need an ISE," he said, emphasizing the ease of creation, the flexibility of the interface, and the strong cross-platform implementation of Web services as key to its success. His overall point was that the ease of creation will be driven not so much by the underlying simplicity of the paradigm, but more by the availability of powerful tools and ISEs for abstracting the details to a simple interface for creation.

Panel Discussion on Tap

On Wednesday afternoon the West Coast Web Services panel discussion took place, another of **SYS-CON**'s energetic CEO panels, a high-octane formula already much enjoyed at the Web Services Edge East event in New York in September.

Moderated by ***Web Services Journal*** editor-in-chief Sean Rhody, the panel consisted of Barry Morris, CEO of IONA Technologies; Greg O'Connor, president of Sonic Software; Dirk Slama, CEO of Shinka Technologies; Eileen Richardson, CEO of Infravio; Ali Kutay, CEO of AltoWeb; and Annrai O'Toole, executive chairman of Cape Clear Software.

Sean Rhody opened the discussion with the same question that opened the East Coast panel: "What is a Web service?" and the panel soon moved on to discuss the role of Web services in the industry.

"You're going to see the power structures in our industry change," asserted Morris, a viewpoint endorsed by O'Toole, who was also adamant that Web services represents a "fundamentally different architecture" than anything we have seen to date.

The point of most controversy was whether Web services represented an evolution or a revolution. Dirk Slama's endorsement of the "evolutionary" view was strongly contested by Morris, who time and again stressed to his fellow panelists and to the entire audience of developers and *i*-technology professionals that there has never been anything like Web services before, that it's a complete and utter revolution.

Morris spoke of Web services as a "flattening" of information infrastructures, such that "more people are able to get at your enterprise value" than ever before. Exposing that enterprise value as Web services is going to be the key to the next phase of e-business, he explained. Just as people writing in Word or doing a presentation in PowerPoint are in effect programming, they are empowered by technology to do something previously doable only by programmers, so, continued Morris, Web services is poised to enable a mass adoption of distributed computing techniques previously available only to software engineers.

Controversy on the Floor

AltoWeb's Ali Kutay took issue with this point of view. He couldn't see how Web services would ever supersede the need for, say, J2EE, since at the enterprise-level there were issues of transaction management and security that Web services simply didn't address adequately. In Kutay's view, "Web services are complementary to what J2EE provides," but they don't, he felt, represent the "revolution" that Morris was arguing they do.

At one point it was suggested that a Web service should use UDDI and WSDL over

Exhibit booths drew well-informed attendees.

CEO panel discussion engrosses audience.

Annrai O'Toole, Executive Chairman, Cape Clear

HTTP, but this caused a fairly significant uproar among the panel members. Barry Morris dismissed UDDI and WSDL, insisting that focusing on the "plumbing" of Web services was missing the point of Web services entirely: Web services is a paradigm that enables corporate developers to finally unite their applications.

Probably the most interesting aspect of the discussion was the completely divergent opinions on the target audience for Web services. Annrai O'Toole and several other members clearly felt that Web services should be ubiquitous and purposefully easy to use for everyone – basically the next HTML. Morris and O'Connor disagreed completely, consigning Web services to the realm of professional developers and programmers.

On Thursday, October 25, Annrai O'Toole gave a keynote in which he set out to prove to delegates that Web services "transforms the whole economics and technology of application integration."

O'Toole outlined to the software engineers, *i*-technology professionals, and senior IS managers in the audience how Web services is encouraging a welcome shift in the competitive strategies used in the computer industry.

"Throughout the past 20 years, the industry has been divided by a series of technological hardware and software platform battles," he explained. "Now, with Web services, the industry has for the first time agreed on ubiquitous interoperability standards, thanks to industry standards such as XML, SOAP, WSDL, and UDDI, offering a solution to the bickering and posturing that has created 20 years of IT incompatibility.

"The arrival of widely accepted standards such as XML and SOAP provides a common base platform," he continued, "that supersedes arguments over operating systems, languages, tools, and applications. It prepares the ground for a new era of cooperation – a Third Way, if you will."

For O'Toole the joy of Web services is that there's nothing actually new in them, "but they're repackaged so that the mainstream developer is reenfranchised," he explained, "putting power back in the hands of the type of developer who would use, say, Visual Basic rather than J2EE."

Known for two decades in the industry for his forthright views, O'Toole was not afraid to expand on this view of how, with the move toward a Web services paradigm, developers and users are back in charge, instead of vendors.

The business impact of Web services on application integration would follow the 80/20 rule, O'Toole made clear. "Eighty percent of the integration that end users will want to do can be delivered by Web services, while 20% will continue to be high-end integration solutions that can't be supplied by Web services." That 80% figure, he said, means that "many businesses that don't do integration at all today, the small and medium enterprises, are soon going to find that with Web services integration is possible – for under $100,000 rather than the $500,000 associated with high-end integration solutions, which often need an additional $1 million to $2 million of consultancy fees on top of that."

O'Toole closed his keynote address by giving a demo of a simple Web service, including publishing it into UDDI. He successfully created it in seven minutes instead of the week he claimed would have been necessary before the introduction of the open standards that are at the heart of the new Web services paradigm

Later that day software engineers and business applications developers enjoyed the benefit of a keynote speaker's 18 years of industry experience in building software tools when they heard Dave Chappell, Sonic Software's VP and chief technology evangelist, deliver the afternoon keynote.

The Importance of Messaging

His keynote, "Web Services Meets Reliability," was well received by the audience, whose own industry experience enabled them to easily follow the technical details of Chappell's contention that messaging in general, and asynchronous messaging in particular, is critically important.

Dave Chappell

Chappell opened with a strong assertion: "In distributed application communications among enterprises and across business entities, the use of SOAP over vanilla HTTP just doesn't cut it. Parties are often unreachable." This gap, Chappell predicted, will be filled in the future by more powerful JAXM messaging providers based on protocols that provide a simple, robust way of addressing those issues – such as JMS.

After explaining how JAXM – the Java API for XML Messaging – can be used for constructing and deconstructing SOAP messages intended to be sent over the wire, Chappell focused on XML messaging and talked the audience through a generic Web services usage model. He actually gave a live demo showing Apache SOAP, Tomcat, and SonicXQ all working together in a Web services interaction, then demonstrated the reliability of the system by shutting down parts of the system and bringing them back up.

He then discussed Web services from several viewpoints, leveraging the views of industry experts to prove the need for Web services.

By the end of the keynote the audience had no doubt that bringing reliability to the Web services infrastructure was an important goal and that messaging will play a key role since, as Chappell put it, "interfaces will always need data transformation and end-to-end guaranteed delivery, along with all the other enterprise needs such as scalability and security." And XML messaging – whether through SOAP or through more rigid sets of rules such as ebXML, JAXM, or JMS – is, within a Web services architecture, the answer.

Ajit Sagar interviews Ben Brauer, Product Marketing Manager, HP (top), and Dirk Slama, CEO, Shinka Technologies (bottom) for SYS-CON Radio.

Exhibitors demonstrate the lastest technologies.

SHOP ONLINE AT **JDJSTORE.COM** FOR BEST PRICES OR CALL YOUR ORDER IN AT **1-888-303-JAVA**

BUY THOUSANDS OF PRODUCTS AT GUARANTEED LOWEST PRICES!

JDJStore.COM

JDJ EDITOR'S CHOICE AWARD · JDJ WORLD CLASS AWARD · JDJ READERS' CHOICE AWARD

GUARANTEED BEST PRICES FOR ALL YOUR JAVA-RELATED SOFTWARE NEEDS

SYBASE
SQL Anywhere Studio v7.0

Sybase® SQL Anywhere Studio is a comprehensive package that provides data management and enterprise synchronization to enable the rapid development and deployment of distributed e-business solutions. Optimized for workgroups, laptops, handheld devices and intelligent appliances, SQL Anywhere extends the reach of a corporation's e-business information to anywhere business transactions occur.

SQL Anywhere Studio (base w/ 1 user) v7.0 . . . . . $\$348^{99}$

SITRAKA
JClass Enterprise Suite v 5.0 Bytecode/Gold Support

Fully scalable to mission-critical development environments, this award-winning suite provides a wide range of high-value GUI functionality including: Tables and grids charting and graphing layout and reporting hierarchical data display data connectivity GUI enhancements data input and validation JAR optimization Everything the professional Java developer needs to build powerful and flexible application front ends is here. Plus, JClass components offer unrivaled IDE, JDK and platform support, and include one year of premium technical support and free upgrades.

JClass Enterprise Suite v 5.0 . . . . $\$2945^{00}$

SITRAKA
JClass Chart 5.0 with Gold Support including Bytecode

JClass Chart gives you the power to create sophisticated, interactive graphs and charts quickly and easily. This data-aware Java component supports many popular types of business and scientific charts, which can be populated with data from a variety of sources including XML.

JClass Chart 5.0 Bytecode/ Gold Support . . . . $\$1234^{00}$

MICROTEK LAB
Scanmaker 4600 with MN100 Camera Bundle

Internet ready scanner for home or office. 3 Easy Touch Buttons for scan, copy and e-mail provide real convenience. With 2400x1200dpi and true 42-bit color, this scanner provides razor-sharp detail and richly saturated colors. Combination digital camera/PC camera included.

Specifications:

Interface Type USB, Platform Windows and Macintosh, Included Hardware USB Cable, Power Adapter, camera.

Scanmaker 4600 w/ MN100 Camera Bundle . . . . $\$164^{99}$

INTERLINK
RemotePoint RF Wireless Handheld w/ Software

No software required for mouse and laser functions! -Omni-directional control from up to 100' away! -Dedicated slide forward & back buttons -Presentation software effects -Launch programs -Zoom or spotlight images on the screen -Hide & reveal slides -On-screen keyboard -Many more!

RemotePoint RF Wireless handheld w/ Software. . . . . $\$169^{99}$

POINTBASE
Mobile Edition 3.5

PointBase Managing data anywhere, anytime

PointBase Mobile Edition is a powerful, 100% Pure Java˙ object-relational database specifically designed for mobile client applications, Internet appliances and wireless devices. PointBase's revolutionary distributed data management delivers the comprehensive capabilities needed to enable these systems, including ultra-small footprint, extensibility, integration with enterprise databases and an entirely new level of self-management and usability features.

PointBase Mobile Edition 3.5. . . . . $\$194^{99}$

JASC
Quickview Plus v6.0

Now you CAN open virtually any file and e-mail attachment with Quick View Plus - the easy way to view virtually any file. Saves money and time - views files from programs you don't have installed on your computer. With just a mouse click, Quick View Plus gives you instant access to over 200 Windows, DOS, Macintosh, and Internet file types. That's more file formats than any other file viewing utility!

Quickview Plus v6.0 . . . . $\$45^{99}$

MACROMEDIA®
JRun™ Server 3.0/3.1 Enterprise 2 CPU Licenses

JRun™ 3.0/3.1 is an easy-to-use J2EE application server and integrated development environment for building and deploying server-side Java applications. From e-commerce to business automation, JRun is the easiest way for developers to deliver advanced business systems faster and at a lower cost than you'd ever thought possible.

JRun Server 3.0/3.1 Enterprise (2 CPU Licenses) $\$8602^{99}$

SEIKO
SmartPad SP-580 Notepad for Palm Organizer

Using infrared technology, the Seiko SmartPad transfers handwritten notes and drawings to Palm organizers. The SmartPad's black leather portfolio resembles a standard executive accessory, but it has a digitizer tablet concealed beneath a 5-by-8-inch note pad on the right side that receives transmissions from the included SmartPen and sends them through the portfolio's imbedded infrared transmitter to a PDA mounted on its left side.

SmartPad SP-580 Notepad for Palm Organizer . . . $\$149^{99}$

SMC
Wireless 11 Mbps Access Point with PC Card

The wireless LAN solution is the easiest alternative to a traditional wired network. It connects instantly with an existing Ethernet installation to support mobile users, temporary work sites, and other applications ranging from the classroom to the boardroom. Wireless technology provides the maximum user mobility, simple and flexible installation options, a reduced cost of ownership (no cabling costs or maintenance) and excellent scalability in supporting network growth. True 11 Mbps wireless is ideal for use with cable modems, DSL or SOHO applications.

Wireless 11 MBPS Access Point . . . . $\$339^{99}$

LINKSYS
WAP11 Wireless Network Access Point

The Instant Wireless Network Access Point from Linksys delivers the freedom to configure your network your way. Utilization of "state-of-the-art" wireless technology gives you the ability to set up workstations in ways you never thought possible; no cables to install means less expense and less hassle. The Instant Wireless Access Point's high-powered antenna offers a range of operation of up to 800 feet, providing seamless roaming throughout your wireless LAN infrastructure; an advanced user authentication feature ensures a high level of network security.

WAP11 Wireless Network Access Point . . . . $\$174^{99}$

ARCHOS
Portable 6GB MP3 Player/USB

Enjoy listening to over 100 hours/6000 minutes of CD-quality music with your ARCHOS Jukebox 6000 MP3 Player/USB Hard Drive. You can conveniently store all your personal files, along with your favorite music selections, and listen to music for 8 hours before recharging the 4 NIMH batteries.Package Includes: Jukebox 6000, 4 Rechargeable Nimh Batteries, MusicMatch Software, USB Interface, A/C Adapter, Stereo Headphones and Pouch.

Portable 6GB MP3 Player/USB . . . . $\$268^{99}$

WWW.JDJSTORE.COM

[WRITTEN BY Roy Hoobler]

Just what are XSLT extensions? From the W3C...

XSL Extensions and Java...

The element extension mechanism allows namespaces to be designated as extension namespaces. When a namespace is thus designated and an element with a name from that namespace occurs in a template, the element is treated as an instruction rather than as a literal result element. The namespace determines the semantics of the instruction.

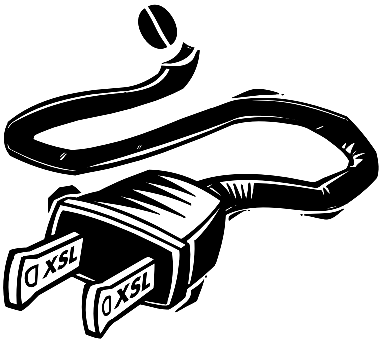
Most people understand that XML is extensible using the extensible Document Object Model. However, eXtensible Stylesheet Language Transformations are also extensible using namespaces. Most functions in XSLT use the XSL namespace that's built in with any parser (such as the xsl function "substring()"). The XSLT specification describes extensions that can extend the functionality in an XSL document.

Following these specifications, you can create your own functions within your own extension namespace (such as <myNameSpace:myFunction...>). The namespace myNameSpace could be a specification, JavaScript, or Java class. Extension namespaces contain functions (such as <xsl:select value-of...>) that return strings or process information and/or provide some other output that you need to include in your document. XML is also extensible using namespaces (which is a big part of how SOAP is implemented), but this is beyond the scope of this article.

In my case we wanted to get a list of JPEG file names from a particular directory and display them as thumbnails in an HTML table. At first we created a Java class that filled this requirement, but some of our application would be in JSP/Java classes and some in XML/XSL. The main problem with this approach is that formatting the page would be in two places and it would be very easy to have malformed HTML. The JSP code would contain some HTML and the XSL sheet would also control parts of the page. I proposed that we include all formatting and HTML within the XSL document to control the display and layout of the page. This is what XSL documents were designed for, right? However, to display the thumbnails, we'd have to call our Java class from inside the XSL document. Figure 1 is our Web page, completely generated via XSL.

Implementing Java Classes Within XSL

I was lucky and got it all to work well the first time. However, trying on three different Web servers (one Tomcat and two JRun servers) took a better understanding of the technology.



The first step is to download Xalan 1.2.2 (Xalan 1) from html:\\xml.apache.org and put the Xalan.jar, Xerces.jar, and BSF.jar into your CLASSPATH. Xalan 2 should work as well; I tried it before on another project, but had problems with the DOM object changes. However, since this application is only doing transformations, Xalan 2 should work fine. A JS.jar file included with the Xalan distribution allows JavaScript functions to be embedded within XSL sheets. You definitely don't need this file if using Java classes.

On Tomcat, each .jar file must be in the tomcat_home\lib or your Java CLASSPATH string under the account that's running Tomcat. I used my /programs/java/lib directory. After you copy the files in here and modify your CLASSPATH, restart Tomcat. You may need to log off or reboot your machine to apply the changes in the CLASSPATH.

With JRun, these .jar files can be in your server/webapps/myapp/WEBINF/-classes directory. After you do this, restart the Web server.

The next step is to test the XSL document page from Listing 1 (from www.ibiblio.org/xml/ by Elliot Harold), which still needs an XML document to parse against and also needs to be transformed. Using a native Java class and method is a great test to see if the system configuration is set up correctly. If you haven't done this, stop and learn how this is done first! If you get a "File Not Found" or "Extension Function Call Failed" error, make sure the BSF.jar file is in the CLASSPATH and in the correct directory. I strongly recommend creating a Java test application as well.

Getting Your Classes to Work as XSLT Extensions

With JRun, your JSP bean classes will also be available as XSLT extensions. However, with Tomcat, they must be loaded when Tomcat starts, using the Java CLASSPATH (sound familiar?). Again, Elliot's example above is easy to implement because the Java Date class is built into the system. Only the Xalan classes are required to get this to work. In either case, after you have the .jar files in place, don't forget to restart JRun or Tomcat.

To get our classes to work across the wide range of servers, I decided to create a .jar file that would include all of the XSLT extension functions needed for this project. These would then be available for "importing" to our JSP pages, JSP beans, and XSL stylesheets. We still use JSP beans but now must take into account the design of our application when creating a new class. Once the .jar file was created – I named ours xslutils.jar – the file is also placed in the /programs/java/lib directory and added to the CLASSPATH.

Author Bio

Roy Hoobler has been involved with Internet programming since 1995 and developing Web businesses and "niche" market sites since 1996. In 1998 he joined Net@Work Inc. (www.netatwork.com) where he designs (and still codes) business and cutting-edge e-commerce applications using a wide range of technologies, including XML. Roy is MCSD certified.

My XSL sheet looks something like Listing 2. The HTML headings are placed on top, and the category or card types are parsed from the XML document (a sample is found in Listing 3). Finally, myfile:createTable is called, which retrieves an HTML table/list of JPEG files from the appropriate directory. It's a little different from the previous example in that I'm also using an LXLST namespace to specify my class or "component." The XSL document parses an XML file that contains category and subcategory information. We also create an href link that reloads the page, passing what was selected. If a subcategory was selected, its directory is passed to our Java class and the thumbnail table is produced.

Common Error Messages

If you receive an "Extension call/new failed" or "Data before root must be well formed" error, then the BSF.jar file isn't loading properly and is actually trying to display your class rather than executing it. Check again to see whether the date function example still works (Listing 1) and make sure the Xalan and BSF .jar files are in the right place. Go back through the "Getting Your Classes to Work..." section.

If you're getting a "File not found" error, then the XSL parser is just not finding your classes. Check to make sure your classes or .jar files are loading and spelled correctly using a JSP or test application. I actually misspelled my .jar file and spent an hour or so trying to figure out what was going on.

Having no errors is generally a good thing. If your class just doesn't seem to be retrieving any information, one trick is to look at the output using "view source" in your browser; some message or output may be there. If you actually see the HTML tags in your browser, make sure you included the disable-output-escaping = "yes" attribute in the "xsl:value-of select" node. Of course, if you want to view the actual HTML code, set this to "no" and the tags will be displayed.

Conclusion

Near the end of the XSL file (Listing 2) is a call to myfile:createTable that creates the file listing needed. It may seem like a lot of work, but now the entire output is generated by one XSL sheet. The classes give us a lot of reusable functionality (in .jar files), and there is a limited amount of code and HTML in the JSP page; we parse only the query string and then perform the XML transformation (see Listing 4). If users want to add a picture, they just upload a .jpg file to the correct directory. If a user needs an additional category, we can update the XML file and create a new directory without worrying about any code that displays the output or retrieves the file listing. If I want to add something to the page, all I have to do is update the XSL stylesheet. Here I need to be more careful because of the complexity of calling the Java components.

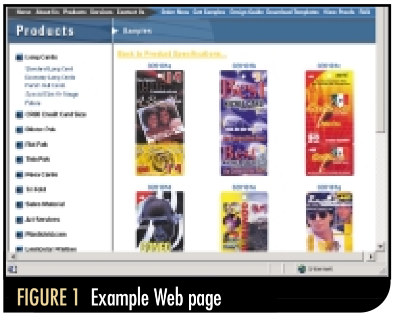


FIGURE 1 Example Web page

References

- ***Apache:*** http:\\xml.apache.org
- ***Mulberrytech XSL Mailing List:*** www.biglist.com/lists/xsl-list/archives/
- ***Cafe con Leche XML News and Resources:*** www.ibiblio.org/xml/
- ***World Wide Web Consortium:*** www.w3.org/

RHOOBLER@NETATWORK.COM

LISTING 1 Testing a server setup for XSLT extensions

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.1"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:date="http://www.cafeconleche.org/ns/">

  <xsl:template match="/">
    <xsl:value-of select="date:new()"/>
  </xsl:template>

  <xsl:script
    implements-prefix="date"
    language="java"
    src="java:java.util.Date"/>

</xsl:stylesheet>
```

LISTING 2 XSL sheet with XSLT extensions

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/
  Transform" xmlns:lxslt="http://xml.apache.org/xslt"
  xmlns:myfile="com.netatwork.webutils.DisplayFiles"
            extension-element-prefixes="myfile"
            version="1.0">

<!-- Default values for our parameters -->
<xsl:param name="requestCategory">lc</xsl:param>
<xsl:param name="directory">lc</xsl:param>
<xsl:template match="/">
   <h1>Hoobler Family Photos...</h1>
<table border="0" cellpadding="4" cellspacing="4"
width="768">
<tr>
    <td valign="top" width="150" class="normal"><br/>
<font face="arial">
You can browse the photos by
clicking on the links below...</font>
      <table border="0" cellpadding="0" cellspacing="0"
        width="165" valign="top">
        <!-- display the main categories -->
        <xsl:for-each select="/thumbnails/category">
        <tr>
<td valign="center" height="30">
 </td>
 <td height="30"><a class="normtextbold">
<xsl:attribute name="href"
>templates.jsp?c=<xsl:value-of
    select="@code" />
</xsl:attribute></a>
            <xsl:value-of select="@name"/></td>
        </tr>
 <xsl:if test="@code[.=$requestCategory]">
 <!-- display the sub categories -->
 <xsl:for-each select="./directory">
 <tr>
<td valign="top"></td>
 <td><img src="images/spacer.gif" width="20" height="2" />
<a class="normblue">
<xsl:attribute name="href">
templates.jsp?c=<xsl:value-of  select="../@code" />
&d=<xsl:value-of select="@path" />
</xsl:attribute>
 <xsl:value-of select="@name"/></a></td>
   </tr>
   </xsl:for-each>
</xsl:if>
</xsl:for-each>
</table>
<br/>
<br/>
</td>
    <td valign="top" width="16" ></td>
 <td valign="top" width="10"></td>
    <td valign="top" width="504" >
<!--Put the @name value of the subcategory (or category
here)-->
<!--if length of the parameter 'directory' is > 0 show
  @name for subdirectory else show the name for the parent
  category -->
<xsl:for-each select="/thumbnails/category[@code=
  $requestCategory]"><br/>
<span class="boldblue"> <xsl:value-of select=
  "@name"/></span><br/><br/>
<span class="normtext"><xsl:value-of select=
  "./description"/><br/></span>
 <xsl:for-each select="./directory[@path=$directory]">
<span class="normtextbold"><xsl:value-of
select="@name"/><br />
 </span>
  </xsl:for-each> <br/>
</xsl:for-each>
<span class="normtext">
<!-- build the table that shows pictures -->
<xsl:variable name="directOut" select=
  "concat('/home/java/jakarta-tomcat-3.2.1/webapps/ROOT/
   templates/',$directory)"/>
<xsl:value-of select="myfile:createTable($directOut,concat
  ('templates/',$directory), 1,'.jpg')"
  disable-output-escaping = "yes" />
</span>
</td>
</tr>
</table>
  </xsl:template>
</xsl:stylesheet>
```

LISTING 3 XML sample data

```
<thumbnails>
<description>
Pictures!
</description>
<category name="Kids" code="kids">
 <description>
  Some pictures of our Kids.
 </description>
 <directory name="Katherine" path="kattie"/>
 <directory name="Michael" path="mike"/>
 <directory name="Joanne" path="joanne"/>
</category>
<category name="Everyone" code="everyone">
 <description>
  Family Shots
 </description>
 <directory name="At Home" path="home"/>
 <directory name="Holidays" path="holiday"/>
</category>
</thumbnails>
```

LISTING 4 XML transformation on a JSP page

```
  strCat = request.getParameter("c").toString();
  strDir = request.getParameter("d").toString();
...
try{
      //The xml bean uses the Xalan parser and does a lot
of work for you!
      objXml.setXMLDoc("products/categoryProductList.xml");
      objXml.setXSLDoc("categoryProductList.xsl");
      objXml.addParameter("requestCategory", strCat);
      objXml.addParameter("directory",strDir);
      objXml.Transform(objPW);
  }catch(Exception e){
  }
```

DOWNLOAD THE CODE @
www.sys-con.com/xml

wireless EDGE
conference&expo

Plan to Exhibit

Provide the Resources To Implement Wireless Strategy

The conference will motivate and educate. The expo is where attendees will want to turn ideas into reality. Be ready to offer solutions.

INTERNATIONAL WIRELESS BUSINESS & TECHNOLOGY CONFERENCE & EXPO

CONFERENCE & EXPO

CONFERENCE & EXPO

Shaping Wireless Strategy for the Enterprise

Santa Clara, CA

May 7-9, 2002

WirelessEdge will provide the depth and breadth of education and product resources to allow companies to shape and implement their wireless strategy. Developers, i-technology professionals and IT/IS management will eagerly attend.

Plan to Attend the 3-DAY Conference

WHO SHOULD ATTEND

Mobile & Wireless Application Professionals who are driving their enterprise's wireless initiatives:

- Program Developers
- Development Managers
- Project Managers
- Project Leaders
- Network Managers
- Senior IT and Business Executives

SHAPE YOUR WIRELESS STRATEGY SAVE THE DATES!

SAVE THE DATES! Wireless Edge May 7-9, 2002
BUSINESS & TECHNOLOGY
wireless
What's the difference between Wireless Learning and Mobile Learning?
Southern California
when incorporating PDAs into the enterprise
So who are you Anyway?
Big Blue Goes Wireless

Conference Tracks

Track One: Development	Track Two: Connectivity	Track Three: Wireless Apps	Track Four: Hardware	Track Five: Business Futures
WAP	Smart Cards	Education	Cell Phones/ WorldPhones	Wireless in Vertical Industries
i-Mode	Wireless LANs incl. Bluetooth	Health Care	PDAs	The WWWW
Bluetooth / 802.11	UMTS/3G Networks	Entertainment	Headphones/ Keyboards / Peripherals	Unwired Management
Short Messaging	Satellite Broadband	Transport	Transmitters/ Base Stations	From 3W to 4W: Issues and Trends
Interactive Gaming		Financial Services	Tablets	"Always-On" Management
GPS / Location-Based		Supply Chain Management		Exploiting the Bandwidth Edge
Wireless Java				Unplugged Valueware
XML & Wireless Technologies				Wireless Sales & Marketing

FOR INFORMATION CALL

201 802-3069

SPEAKER PROPOSALS INVITED

WWW.SYS-CON.COM

EXCLUSIVE SPONSORSHIPS AVAILABLE

Rise above the noise. Establish your company as a market leader. Deliver your message with the marketing support of

SYS-CON MEDIA

SYS-CON EVENTS

THE LARGEST WEST COAST WIRELESS BUSINESS & TECHNOLOGY CONFERENCE OF THE YEAR!

DYNAMIC MULTIMEDIA PRESENTATIONS USING XSLT

HOW TO TRANSFORM STATIC DATA DESCRIPTORS FOR AUDIO AND IMAGES ON THE WEB

WRITTEN BY SIET LENG LAI

SIETLENG_LAI@YAHOO.COM

AUTHOR BIO

Siet Leng Lai has published and presented at several international conferences and is now working on XML, WML, SMIL, VoXML, mobile Internet, and peer-to-peer computing. Since 1990 she has been a lecturer at Ngee Ann Polytechnic, Singapore, on subjects ranging from system programming, networking, and Internet computing to XML, Web development, and parallel and distributed computer systems. She holds degrees from the National University of Singapore.

DYNAMIC MULTIMEDIA PRESENTATIONS USING XSLT

Synchronized Multimedia Integration Language – or SMIL, pronounced *smile* – is an XML application and currently a W3C recommendation. The basic idea of SMIL is to give media components – text, images, audio, video, and the like – a URL and to allow you to control what, where, when, and how the media components appear in a dynamic multimedia presentation.

For instance, you could specify the layout of the presentation (“where”); various media elements like audio or image (“what”); temporal association (“when”), e.g., a text element appears 5 seconds after an image is shown; and the transition between elements (“how”) in your multimedia presentations. There are various potential SMIL applications such as e-learning, entertainment, training programs for corporations or institutions, dynamic product catalog presentations, multimedia slide shows for travel promotions, photograph album slide shows, and so on.

A few SMIL players – such as Oratrix’s GriNs, RealNetwork’s RealPlayer, and Microsoft Internet Explorer 5.5 – are available. IE 5.5 supports many of the SMIL 2.0 draft modules, including Timing and Synchronization, BasicAnimation, SplineAnimation, BasicMedia, MediaClipping, and BasicContentControl.

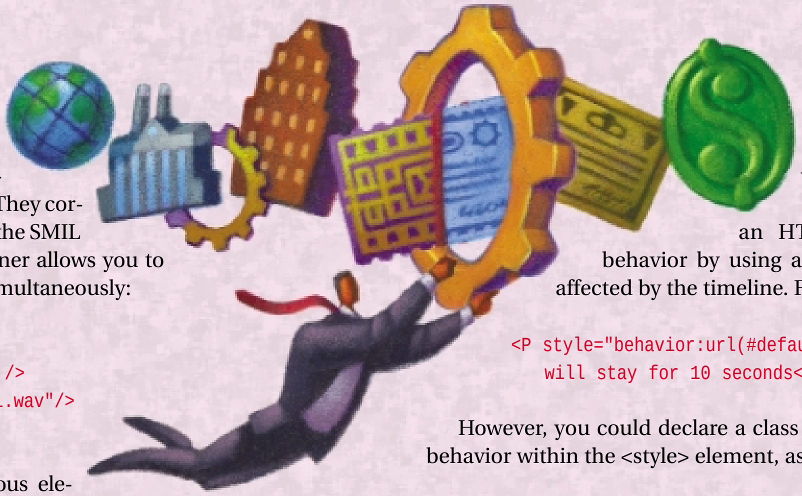
In my previous article, “Multimedia Slide Show on the Web Using SMIL” (***XML-J***, Vol. 2, issue 2), I discussed and demonstrated a multimedia slide show for travel portals using Active Server Pages, a back-end database, and SMIL (using Oratrix’s griNs player) on the Web. In this article I introduce Microsoft’s HTML+TIME, an implementation of the HTML+SMIL profile in SMIL 2.0. I’ll use the travel slide show application from my previous article to further explain how XML and XSLT can be used to transform static data descriptors for audio and images into a dynamic multimedia presentation on the Web. We’ll see in detail how to combine the power of XSLT and HTML+TIME to create and transform data.

Overview of HTML+Time

One of the main design goals of SMIL is to allow reuse of SMIL syntax and semantics in other XML-based languages, especially by those who need to represent timing and synchronization. The HTML+SMIL language profile defined by the SMIL Working Draft specifies that HTML can be used for layout in the HTML+SMIL profile rather than the separate SMIL Language for layout.

The obvious advantage here is that you don’t need to learn a new language. For example, adding HTML+TIME attributes to existing HTML elements would provide timing effects on your Web pages. HTML+TIME is the Microsoft implementation of the HTML+SMIL profile in SMIL 2.0. However, the HTML+SMIL language profile is not included in the SMIL 2.0 recommendation. Instead, it will continue to be updated and published as a separate working draft.

HTML+TIME provides a declarative way to add multimedia and interactivity, timing and media synchronization support, to Web pages. You can add images, streaming video, or sounds to an HTML page and synchronize them with text layout for a specified duration. HTML elements behave differently when you add time attributes, depending on the HTML element type. I’ll discuss the essential HTML+TIME elements used in this article only briefly here. The Microsoft Web site contains more detailed information (see **References** section).



This article uses two important time containers in the HTML+TIME elements for synchronization: t:PAR and t:SEQ. They correspond to the PAR and SEQ in the SMIL specification. The t:PAR container allows you to group elements to be played simultaneously:

```
<t:PAR>
    <t:IMG  SRC="japan1.jpg" />
    <t:AUDIO  SRC="narration1.wav"/>
</t:PAR>
```

HTML+TIME provides various elements for playing back different media objects, such as video, audio, and images. The <t:AUDIO> and <t:IMG> elements are used to play back sound clips and still images. As shown in the code segment above, the image object accompanied by the audio track will be played simultaneously.

The t:SEQ container allows you to play multiple elements in sequence:

```
<t:SEQ>
    <t:IMG SRC="japan1.jpg" dur="5" />
    <t:IMG SRC="japan2.jpg" dur="8" />
</t:SEQ>
```

With reference to the above code segment, the image "japan1.jpg" remains visible for 5 seconds, followed by the image "japan2.jpg" shown for 8 seconds. Here we use the simple duration attribute "dur" to specify the amount of time that the element remains active.

To use HTML+TIME elements such as t:PAR or t:SEQ, you first have to declare the XML namespace t: in HTML tags:

```
<HTML XMLNS:t ="urn:schemas-microsoft-com:time">
```

To establish t: as the namespace, you must import the HTML+TIME version 2 behavior into the namespace using the "import" directive as follows:

```
<?IMPORT namespace="t" implementation="#default#time2">
```

HTML+TIME is implemented as a DHTML default behavior, one of the powerful new features introduced in IE 5. DHTML behaviors are simple lightweight components that enhance the default behavior of a standard HTML element. In fact, behaviors are also a current working draft on the W3C site.

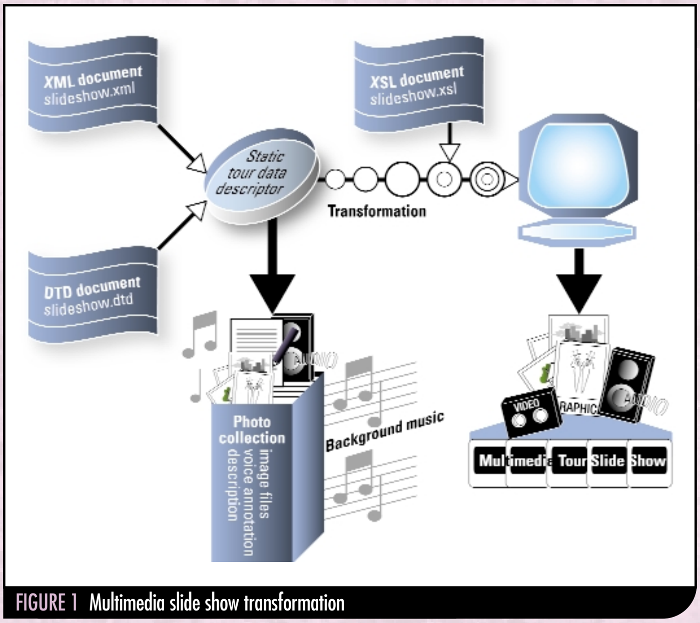


FIGURE 1 Multimedia slide show transformation

In addition, you may associate an HTML element with the time2 behavior by using an inline style so the element is affected by the timeline. For instance:

```
<P style="behavior:url(#default#time2)" dur="10"> This line
   will stay for 10 seconds</P>
```

However, you could declare a class for the HTML+TIME version 2.0 behavior within the <style> element, as follows:

```
<style>
   .time { behavior: url(#default#TIME2); }
</style>
```

You could then reuse the declaration for any of the intended HTML elements, as shown below:

```
<P class="time" dur="10"> This line will stay for 10 seconds</P>
```

Listing 1 is a simple outline of a Web page with HTML+TIME elements.

Anatomy of a Slide Show Transformation

A multimedia slide show is a simple yet effective means of providing a virtual tour so you can view animated, precaptured scenery with the appropriate narration. It provides a rather convenient, low-cost interactive solution without demanding too much bandwidth as compared with video. With the growing popularity of digital cameras, taking a snapshot of a still image in digital format is rather easy. Although it doesn't provide the continuity of video, it's good enough to be used as a virtual tour for promotion purposes, especially as it has slide show facility.

It actually provides a useful service to travel portals. In my previous article I demonstrated the potential of SMIL in multimedia presentations and how it can be used by the travel industry as a value-added service to travel portal sites. I'll further illustrate how XML and XSLT can be used to transform static descriptor data into the SMIL-based (using HTML+TIME) multimedia slide show shown in Figure 1. The HTML+TIME implementation doesn't require any special plug-in player as the IE 5.5 browser already supports it.

Figure 2 shows the hierarchy of the corresponding XML document in Listing 2 for the virtual tour slide show. It consists of the static descriptor for the tour data, including the name and description of the tour and the photo collection for each slide. The photo collection includes the title of the slide, the image snapshot, and audio narration.

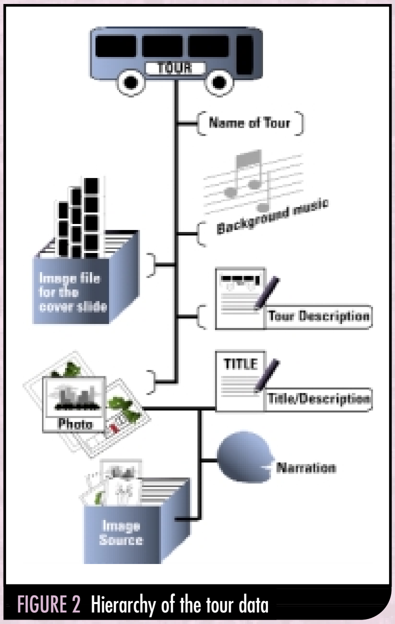


FIGURE 2 Hierarchy of the tour data

Listing 3 shows the Document Type Definition for the XML document. Listing 4 shows the extensible stylesheet to transform the static descriptor to the dynamic HTML+TIME Web page. Figure 3 shows the outcome of the transformation. You need to have an IE version 5.5 browser (required for HTML +TIME version 2), and you also need to ensure that your XML parser is updated to MSXML version 3.0 in order to support

Missed an issue?

We've got 'em all for you on CD!

JAVA DEVELOPER'S JOURNAL

The most complete library of exclusive *JDJ* articles on one CD!

Check out over 500 articles covering topics such as...
Java Fundamentals, Advanced Java, Object Orientation, Java Applets, AWT, Swing, Threads, JavaBeans, Java & Databases, Security, Client/Server, Java Servlets, Server Side, Enterprise Java, Java Native Interface, CORBA, Libraries, Embedded Java, XML, Wireless, IDEs, and much more!

***JDJ* The Complete Works**
~~Reg $119.99~~
Buy Online
Only $71.99

XML JOURNAL

The most complete library of exclusive *XML-J* articles on one CD!

Check out over 150 articles covering topics such as...
XML in Transit, XML B2B, Java & XML, The XML Files, XML & WML, Voice XML, SYS-CON Radio, XML & XSLT, XML & XSL, XML & XHTML, 2B or Not 2B, XML Industry Insider, XML Script, <e-BizML>, XML & Business, XML Demystified, XML & E-Commerce, XML Middleware, and much more!

***XML-J* The Complete Works**
~~Reg $59.99~~
Buy Online
Only $53.99

COLDFUSION Developer's Journal

The most complete library of exclusive *CFDJ* articles!

Check out over 250 articles covering topics such as...
Custom Tags, ColdFusion and Java, Finding a Web Host, Conference Reports, Server Stability, Site Performance, SYS-CON Radio, ColdFusion Tips and Techniques, Using XML and XSLT with ColdFusion, Fusebox, Building E-Business Apps, Applicaton Frameworks, Error Handling, and more!

***CFDJ* The Complete Works**
~~Reg $79.99~~
Buy Online
Only $71.99

SPECIAL OFFER:
Buy CFDJ & CFA The Complete Works
For Only $129.99

CF Advisor

The most complete library of exclusive *CFA* articles!

Check out over 200 articles covering topics such as...
E-Commerce, Interviews, Custom Tags, Fusebox, Editorials, Databases, News, CF & Java, CFBasics, Reviews, Scalability, Enterprise CF, CF Applications, CF Tips & Techniques, Programming Techniques, Forms, Object-Oriented CF, WDDX, Upgrading CF, Programming Tips, Wireless, Verity, Source Code, and more!

***CFA* The Complete Works**
~~Reg $79.99~~
Buy Online
Only $71.99

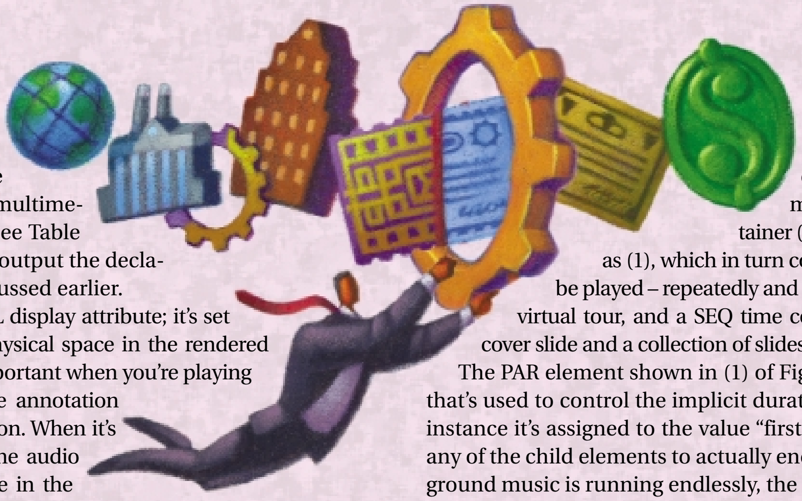
JDJStore.com

Order Online and Save 10% or More!

www.JDJSTORE.com

OFFER SUBJECT TO CHANGE WITHOUT NOTICE

COLLECT ALL 4
FOR ONLY $229.99
JDJ, XML-J
CFDJ & CFA



the proper XSL transformation of the given codes.

Now let's scrutinize the transformation detail for the multimedia slide show. Lines 5 to 13 (see Table 1) of the stylesheet in Listing 4 output the declaration for HTML+TIME as discussed earlier.

Line 10 declares the DHTML display attribute; it's set to "none" so it won't reserve physical space in the rendered document. This is especially important when you're playing the background music or voice annotation together with the photo collection. When it's being played, you don't want the audio media element to occupy space in the rendered document. You can see from Listing 4 that the disp class is applied to the <t:audio> element.

Line 12 is needed to import the time2 behavior into t: namespace, as mentioned earlier. We need to control the output escaping from the import directive; otherwise it will be taken as an XML processing instruction. Here I use <xsl:text> to output the literal text to the current output destination with the attribute "disable-output-escaping" set to "yes" so the character enclosed will be output as it is. For instance, "<" will be output as it is and it will not be escaped using the normal XML escaping mechanisms.

To help you understand the transformation and the target output, I've included the generated output in Listing 5 and an outline of the generated document in Figure 4 for illustration purposes.

Line 5	<html xmlns:t="urn:schemas-microsoft-com:time">
6	<head>
7	<title> Virtual Tour Slide Show </title>
8	<style>
9	.time { behavior: url(#default#time2); }
10	.disp { display: "none" }
11	</style>
12	<xsl:text disable-output-escaping="yes">&lt;?IMPORT namespace="t" implementation="#default#time2"&gt;</xsl:text>
13	</head>

TABLE 1 Header section for the stylesheet slideshow.xsl

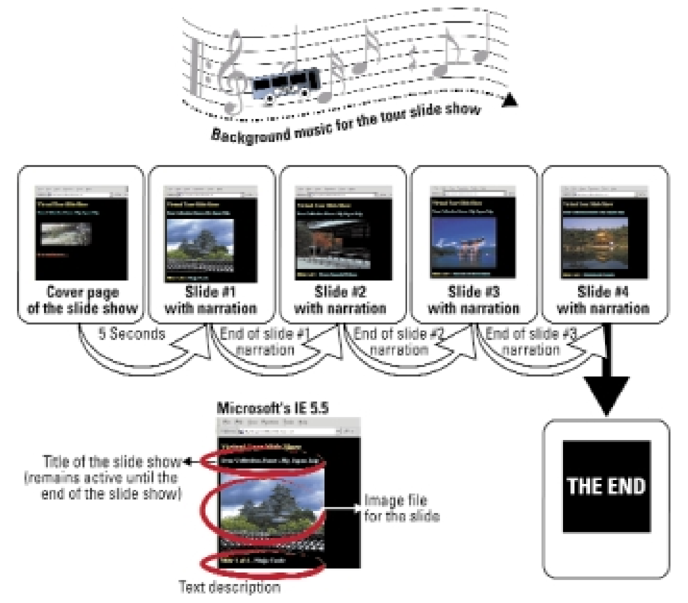

Background music for the tour slide show
Cover page of the slide show
Slide #1 with narration
Slide #2 with narration
Slide #3 with narration
Slide #4 with narration
5 Seconds
End of slide #1 narration
End of slide #2 narration
End of slide #3 narration
Microsoft's IE 5.5
Title of the slide show (remains active until the end of the slide show)
Image file for the slide
Text description
THE END

FIGURE 3 Virtual tour slide show with XSLT transformation

First let's examine the generated output document. With reference to Figure 4, the target document consists mainly of a PAR container (in an initial SEQ container) labeled as (1), which in turn consists of the background music to be played – repeatedly and forever – the title text object for the virtual tour, and a SEQ time container. The latter comprises the cover slide and a collection of slides presenting the tour.

The PAR element shown in (1) of Figure 4 has an "endsync" attribute that's used to control the implicit duration of the time container. In this instance it's assigned to the value "first." This means that it will wait for any of the child elements to actually end its active duration. As the background music is running endlessly, the PAR element will complete when the slide presentation ends its active duration.

Label (2) in Figure 4 indicates that the <t:audio> element for the background music has an ID, namely "bg", which will be used for subsequent referencing. The element also has a "dur" attribute set to "indefinite" as the background music is supposed to be played throughout the slide show presentation.

The SEQ element shown in label (3) of Figure 4 has a begin attribute that's set to "bg.begin" where "bg" is the ID of the background music. In other words, it will start only when the background music starts. The cover slide with the cover image file will be displayed for 3 seconds – see label (4) of Figure 4 – before starting the slide presentation. The slide presentation is represented as a sequential collection of photographs. Each photo comprises the corresponding image file, for example, japan1.gif; voice annotation, that is, the narration of the photo; and the slide title/description. All these elements within a slide are presented in parallel.

Label (5) of Figure 4 shows one of the photos in the collection. The PAR container will end when the narration ends its active duration. It is achieved by using the "endsyn" attribute. As shown in the figure, the "endsyn" attribute is assigned to the ID of the corresponding voice narration, such as "audio1", and the corresponding voice narration – see label (6) in Figure 4 – has an ID attribute set to "audio1". Upon completion of the PAR element, it will proceed to the next photo scene until the end of the sequence.

With reference to the following code segment in the extensible stylesheet (refer to Listing 4):

```
<xsl:for-each select="Tour">
  …
  <xsl:variable name="total"  select="count(Photo)" />
```

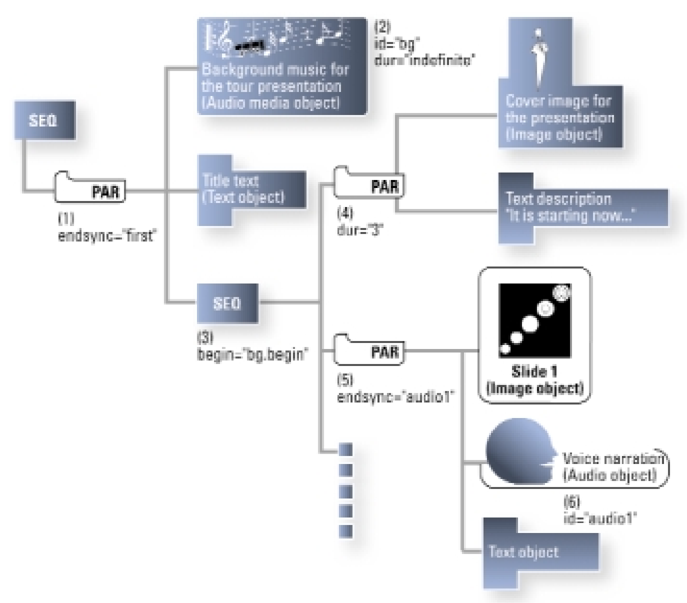

SEQ
PAR
(1) endsync="first"
Background music for the tour presentation (Audio media object)
(2) id="bg" dur="indefinite"
Title text (Text object)
PAR
(4) dur="3"
Cover image for the presentation (Image object)
Text description "It is starting now..."
SEQ
(3) begin="bg.begin"
PAR
(5) endsync="audio1"
Slide 1 (Image object)
Voice narration (Audio object)
(6) id="audio1"
Text object

FIGURE 4 Outline of transformed document with HTML+TIME

Upon selecting "Tour" nodes, you compute the total number of Photo nodes in the Tour node. This indicates the total number of slides to be shown.

```
<xsl:for-each select="Photo">
   …
      <xsl:variable name="syn" select="concat('audio', position())" />

      <t:par  endsync="{$syn}">
              <t:img  … >
               <t:audio  id="{$syn}"  … />
                   …
               <SPAN class="time …>
                     …
                   Slide <xsl:value-of select="position()" /> of
<xsl:value-of select="$total" /> -
                    …
                </SPAN>
```

For each of the Photo nodes, you derive the ID of the <t:audio> element for voice narration by concatenating the string "audio" and the number currently assigned to the current node in the list. The first Photo node, for example, is given "audio1". This is used to synchronize the still image and the audio narration, as discussed previously. In addition, you'll display the slide sequence in terms of the current slide position and the total number of slides.

You can also use server-side scripting (e.g., ASP, shown in Listing 6, to invoke the transformation. With reference to this listing, you first create instances of document objects, that is, xmldoc and xsldoc, using the Microsoft XMLDOM parser. You then load, respectively, the XML document – slideshow.xml – and the XSL document – slideshow.xsl – into memory. The last line of Listing 6 transforms the XML document using the XSL document. It then returns the outcome to the browser. Upon successful transformation (see Figure 3), the specified music plays continuously in the background, and each slide will transit to another only when the respective voice narration ends. This is a cost-effective means to provide dynamic visual tour information to users on the Web.

Summary

This article has discussed in detail a virtual tour slide show application that combines the power of Microsoft SMIL implementation (HTML+TIME) and XSLT transformation. The use of SMIL for multimedia presentations on the Web is very promising. With the commitment to SMIL by various major industrial partners such as Microsoft, and as more SMIL-based applications are introduced, we hope to see greater acceptance and implementation of SMIL in the future.

References

- *W3C synchronized multimedia SMIL 2.0 specification:* www.w3.org/TR/smil20/
- *Microsoft HTML+TIME:* http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dntime/html/htmltime.asp
- *HTML+SMIL language profile:* http://www.w3.org/TR/2001/WD XHTMLplusSMIL-20010807/
- *DHTML behaviors:* http://msdn.microsoft.com/library/default.asp?url=/workshop/author/behaviors/overview.asp
- *W3C Behavioral Extensions to CSS:* www.w3.org/TR/becss

LISTING 1

```
<HTML XMLNS:t ="urn:schemas-microsoft-com:time">
<HEAD>
<STYLE>
   .time {behavior: url(#default#time2);}
</STYLE>
<?IMPORT namespace="t" implementation="#default#time2">
</HEAD>
<BODY>
<t:SEQ>
<SPAN CLASS="time" DUR="2">
         <h1>Introduction to HTML+Time version 2.0 </h1>
</SPAN>
…
<t:IMG … >
…
</t:SEQ>
</BODY>
</HTML>
```

LISTING 2

```
<?xml version="1.0"?>
<!-- this contains Tour information -->
<?xml:stylesheet type="text/xsl"
             href="slideshow.xsl"?>
<!DOCTYPE Tour SYSTEM "slideshow.dtd">
<Tour>
   <name>My Japan Trip</name>
   <background>mymusic.mid</background>
   <cover>japan0.jpg</cover>
   <description>This trip provides quick glimpse
           on the scenery, their culture
          and lifestyles</description>
   <Photo id="1">
      <Title>Ninjo Castle</Title>
      <Narration>japan1.wav</Narration>
      <Source>japan1.gif</Source>
   </Photo>
   <Photo id="2">
      <Title>Kyoto Imperial Palace</Title>
      <Narration>japan2.wav</Narration>
      <Source>japan2.gif</Source>
   </Photo>
   <Photo id="3">
      <Title>Pine Isle of Matsushima</Title>
      <Narration>japan3.wav</Narration>
      <Source>japan3.gif</Source>
   </Photo>
  <Photo id="4">
      <Title>Kinkakushi Temple</Title>
      <Narration>japan4.wav</Narration>
      <Source>japan4.gif</Source>
   </Photo>
 </Tour>
```

LISTING 3

```
<!ELEMENT Tour (name, background, cover,
                  description, Photo*)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT background (#PCDATA)>
<!ELEMENT cover (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT Photo (Title, Narration, Source)>
<!ATTLIST Photo id CDATA #REQUIRED>
<!ELEMENT Title (#PCDATA)>
<!ELEMENT Narration (#PCDATA)>
<!ELEMENT Source (#PCDATA)>
```

LISTING 4

```
<?xml version='1.0'?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output  indent="yes"/>
<xsl:template match="/">
<html xmlns:t="urn:schemas-microsoft-com:time">
  <head>
  <title> Virtual Tour Slide Show </title>
  <style>
    .time    { behavior: url(#default#time2); }
    .disp    { display: "none" }
 </style>
  <xsl:text disable-output-escaping="yes">
      &lt;?IMPORT namespace="t"
       implementation="#default#time2"&gt;
  </xsl:text>

  </head>
  <body bgcolor="black">
  <t:seq>
   <xsl:for-each select="Tour">
       <xsl:variable name="coverimage"
```

```
                       select="cover" />
       <xsl:variable name="bgmusic"
                       select="background" />
       <xsl:variable name="total"
                       select="count(Photo)" />

       <t:par endsync="first">
       <t:audio id="bg" timeAction="class:disp"
              src="{$bgmusic}"  begin="0"
              dur="indefinite" />

        <SPAN class="time"
                      timeAction="display"
                      begin="bg.begin">
              <font color="yellow">
               <h2>Virtual Tour Slide Show</h2>
              </font>
              <font color="cyan">
               <h3>
                Tour Collection Name :
                <xsl:value-of select="name" />
               </h3>
               </font>
         </SPAN>


          <t:seq begin="bg.begin">
              <t:par dur="3">
                  <t:img  timeAction="display"
                              src="{$coverimage}" />

                   <SPAN class="time"
                              timeAction="display">
                     <font class="time"
                                color="yellow">
                      <h4>
                        <xsl:value-of
                           select="description" />
                       </h4>
                       </font>
                        <font class="time"
                                   color="red">
                         <h3>
                         It is starting now ...
                         </h3>
                      </font>
                </SPAN>
          </t:par>

    <xsl:for-each select="Photo">

        <xsl:variable name="imagesource"
                                select="Source" />
        <xsl:variable name="narration"
                                select="Narration" />
         <xsl:variable name="title"
                                select="Title" />
         <xsl:variable name="syn"
             select="concat('audio', position())" />

         <t:par  endsync="{$syn}">
              <t:img  timeAction="display"
                            src="{$imagesource}" />

              <t:audio  id="{$syn}"
                   timeAction="class:disp" begin="0"
                        src="{$narration}" />

              <SPAN class="time"
                            timeAction="display">
               <h3>
                    <font color="yellow">
                     Slide
                    <xsl:value-of
                         select="position()" /> of
                    <xsl:value-of
                         select="$total" /> -
                     </font>
                     <font color="cyan">
                        <xsl:value-of
                            select="$title" />
                     </font>
```

```
               </h3>
             </SPAN>

           </t:par>

   </xsl:for-each>

       </t:seq>
      </t:par>
  </xsl:for-each>
  </t:seq>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

LISTING 5

```
<html xmlns:t="urn:schemas-microsoft-com:time">
<head>
<META http-equiv="Content-Type"
              content="text/html; charset=UTF-16">
<title> Virtual Tour Slide Show </title>
<style>
    .time         { behavior: url(#default#time2); }
    .disp    { display: "none" }
 </style>
<?IMPORT namespace="t"
                  implementation="#default#time2">
</head>

<body bgcolor="black">
<t:seq>
   <t:par endsync="first">
        <t:audio id="bg" timeAction="class:disp"
                       src="mymusic.mid"
                       begin="0" dur="indefinite"/>
        <SPAN class="time" timeAction="display"
                      begin="bg.begin">
           <font color="yellow">
           <h2>Virtual Tour Slide Show</h2>
           </font>
           <font color="cyan">
           <h3>Tour Collection Name :
                     My Japan Trip</h3>
           </font>
        </SPAN><t:seq begin="bg.begin">
        <t:par dur="3">
             <t:img timeAction="display"
                        src="japan0.jpg"/>
             <SPAN class="time"
                           timeAction="display">
               <font class="time" color="yellow">
                <h4>This trip provides quick
                 glimpse on the scenery, their
                 culture and lifestyles
                </h4>
                </font>
                <font class="time" color="red">
                <h3>It is starting now ... </h3>
                </font>
             </SPAN>
        </t:par>
        <t:par endsync="audio1">
           <t:img timeAction="display"
                         src="japan1.gif"/>
           <t:audio id="audio1"
                     timeAction="class:disp"
                     begin="0" src="japan1.wav"/>
              <SPAN class="time"
                           timeAction="display">
                     <h3>
                      <font color="yellow">
                       Slide 1 of 4 -
                      </font>
                      <font color="cyan">
                         Ninjo Castle
                      </font>
                     </h3>
              </SPAN>
        </t:par>
        <t:par endsync="audio2">
              <t:img timeAction="display"
```

```
                src="japan2.gif"/>
     <t:audio id="audio2"
            timeAction="class:disp"
            begin="0" src="japan2.wav"/>
     <SPAN class="time"
                   timeAction="display">
          <h3>
            <font color="yellow">
              Slide 2 of 4 -
            </font>
            <font color="cyan">
               Kyoto Imperial Palace
            </font>
          </h3>
      </SPAN>
</t:par>
<t:par endsync="audio3">
        <t:img timeAction="display"
                   src="japan3.gif"/>
        <t:audio id="audio3"
              timeAction="class:disp"
              begin="0" src="japan3.wav"/>
        <SPAN class="time"
                    timeAction="display">
            <h3>
               <font color="yellow">
                  Slide 3 of 4 -
               </font>
                <font color="cyan">
                  Pine Isle of Matsushima
                </font>
            </h3>
       </SPAN>
 </t:par>
 <t:par endsync="audio4">
      <t:img timeAction="display"
                 src="japan4.gif"/>
      <t:audio id="audio4"
              timeAction="class:disp"
              begin="0" src="japan4.wav"/>
```

```
               <SPAN class="time"
                             timeAction="display">
                   <h3>
                      <font color="yellow">
                          Slide 4 of 4 -
                      </font>
                      <font color="cyan">
                         Kinkakushi Temple
                      </font>
                   </h3>
                </SPAN>
           </t:par>
      </t:seq>
   </t:par>
</t:seq>
</body>
</html>
```

LISTING 6

```
<%
dim xsldoc
dim xmldoc

'Load the XML
set xmldoc =
      Server.CreateObject("Microsoft.XMLDOM")
xmldoc.async = "false"
xmldoc.load(Server.MapPath("slideshow.xml"))

'Load the XSL
set xsldoc =
      Server.CreateObject("Microsoft.XMLDOM")
xsldoc.async = false
xsldoc.load(Server.MapPath("slideshow.xsl"))

'Transform the file
response.write(xmldoc.transformNode(xsl))

%>
```

DOWNLOAD THE CODE @
www.sys-con.com/xml

BandWidth
Integrating ColdFusion and Flash
COLDFUSION Developer's Journal
SIEBEL
Multichannel STRATEGIES
TANGO IN A JAVA SERVER WORLD
DEVELOPERS JOURNAL
TANGO
SYBASE

PRODUCT REVIEW

IpedoXML Database 2.0

by Ipedo, Inc.

Author Bio

Jim Milbery, a software consultant with Kuromaku Partners LLC (www.kuromaku.com), based in Easton, Pennsylvania, has over 17 years of experience in application development and relational databases. He is the applications editor for **Wireless Business & Technology***, the product review editor for* **Java Developer's Journal***, and the author of* Making the Technical Sale.

JMILBERY@KUROMAKU.COM

[Reviewed By **Jim Milbery**]

SPECS

Ipedo, Inc.

1001 Marshall Street
Redwood City, CA 94063
Phone: 650 306-4000
Web: www.ipedo.com

Test Environment

OS: Windows 2000
Processor: Toshiba Satellite Pro 4600
Memory: 256MB

Specifications

Platforms: Windows NT/2000, Linux, Solaris
Pricing: $29,000 per CPU for all platforms

The advantages of XML as a data storage and transfer mechanism are well known and almost universally understood. However, much of the data that will be manipulated with XML is currently stored inside back-end systems. XML tends to be more verbose than some other traditional data storage technologies, so it's becoming important for customers to find new ways to speed up the delivery process of XML documents across the enterprise. One solution to this problem is the Ipedo XML database from Ipedo, a Redwood City, California, start-up company.

This native XML database, written in Java, can be deployed alongside your application server to provide a speedy solution for XML processing and XSLT conversions. The Ipedo team has identified three critical growth markets for XML technology:

- *Web applications*
- *Wireless applications*
- *Business-to-business systems*

Personalized Web applications rely on personalization data and content that has traditionally been stored within the file system and relational databases. B2B applications rely on XML for transaction sets and to display customized information for users in a portal context. Wireless applications may be the ultimate test for customization, since these applications need to support personalized content for dozens of heterogeneous devices. The Ipedo XML database 2.0 relies on in-memory technology so it can eliminate the XML processing bottlenecks that are all too common in these three types of applications. It can also act as a cache for XML data at the middle tier (or at the edge of the network) to reduce total round-trip time for transactions. Ipedo 2.0 even supports the Scalable Vector Graphics (SVG) file format within the database, allowing developers to include images inside their XML documents.

Installing and Working with Ipedo

Ipedo provided me with a demonstration copy of the Ipedo server, but I also used their Web site to request a public download. The Windows NT/2000 download kit is packaged inside a 15MB InstallShield setup program. I had no trouble getting the Ipedo software installed and running. The entire process took only a matter of minutes.

In order to work with Ipedo you should orient yourself to the software by using the Ipedo console, which is itself a Java application. Once the software is installed, you can create a new Ipedo XML database using the console. From the console application you can also import schemas, DTDs, and XML documents into the new in-memory database, as shown in Figure 1.

Ipedo provides a complete Java API for their server, so you don't have to rely on the console to work with Ipedo, but the console makes it easier to get yourself oriented to the server. However, I expect that most developers would rely on the API when using Ipedo.

Since the Ipedo technology is designed to run alongside application servers and edge servers, Java is the ideal language to use with software. I can envision dynamic Web application developers using servlets to access XML data stored within Ipedo as part of a highly personalized, dynamic application. However, the Ipedo console application has enough of the features and functions to get you started.

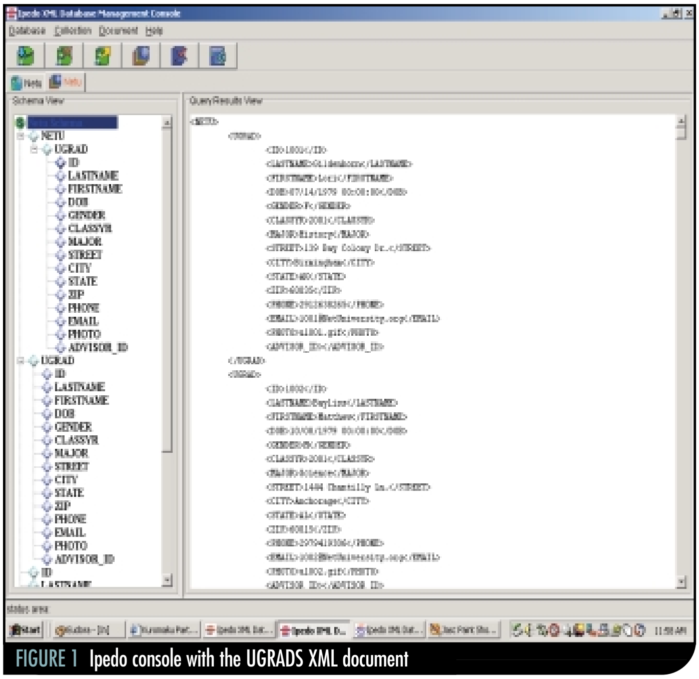
I was able to load my Net University (NETU) DTD and UGRADS XML document in no time. XML documents are stored within "collections" in the Ipedo environment, and the collections list appears in the left-hand panel (see Figure 1). Ipedo offers built-in support for XSLT and XPath/XQuery, and it's a simple process to load up a set of documents and run queries against the data. I entered a simple search for my NETU data in the standard XPath Query format:

```
/NETU/UGRAD[CLASSYR='2002']
```

The format of the search string follows the DTD for your XML documents. In my example case I chose to search within the NETU schema for UGRAD records whose CLASSYR was equal to the value "2002". The console interface provides only very basic access to features and functions, and this is one of the cases in which these limitations are most obvious. XPath queries are case sensitive, but the console doesn't give you any indication that you've entered an invalid query. Although this is an inherent limitation in XPath, the Ipedo console does not offer you any special assistance with the querying process. However, Ipedo has added support for free-form XML search to the database 2.0 release, which will definitely make it easier and faster to find information in complex XML documents.

If you plan on searching across large collections of documents and data, Ipedo can speed up the process even further through the use of user-defined indexes (see Figure 2). I loaded a larger document from the NETU schema and then indexed the STATE field using a B-tree format. Ipedo supports hash maps (which are good for key values) and B-trees, which offer better performance for nonunique values (such as "states"). Once the index was created, I noticed a nice boost in performance against my larger data sets.

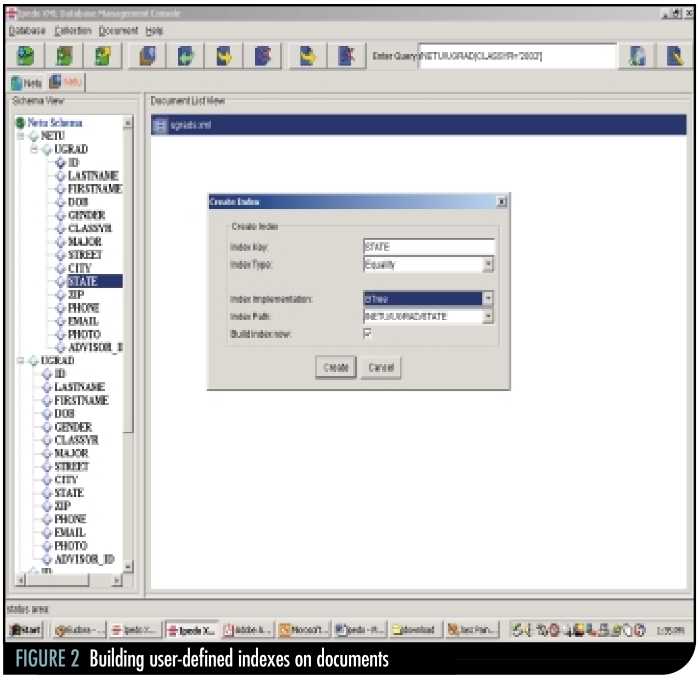
Ipedo provides a detailed reviewer's guide and a very comprehensive set of developer



Ipedo XML Database Management Console
Database Collection Document Help
Schema View
QueryResults View

```
<NETU>
        <UGRAD>
                <ID>1801</ID>
                <LASTNAME>Gildenhorne</LASTNAME>
                <FIRSTNAME>Lori</FIRSTNAME>
                <DOB>07/14/1979 00:00:00</DOB>
                <GENDER>F</GENDER>
                <CLASSYR>2001</CLASSYR>
                <MAJOR>History</MAJOR>
                <STREET>139 Bay Colony Dr.</STREET>
                <CITY>Birmingham</CITY>
                <STATE>AL</STATE>
                <ZIP>60826</ZIP>
                <PHONE>2912638269</PHONE>
                <EMAIL>1801@NetUniversity.org</EMAIL>
                <PHOTO>a1801.gif</PHOTO>
                <ADVISOR_ID></ADVISOR_ID>
        </UGRAD>
        <UGRAD>
                <ID>1802</ID>
                <LASTNAME>Sayline</LASTNAME>
                <FIRSTNAME>Matthew</FIRSTNAME>
                <DOB>10/08/1979 00:00:00</DOB>
                <GENDER>M</GENDER>
                <CLASSYR>2001</CLASSYR>
                <MAJOR>Science</MAJOR>
                <STREET>1444 Chantilly Ln.</STREET>
                <CITY>Anchorage</CITY>
                <STATE>AK</STATE>
                <ZIP>60815</ZIP>
                <PHONE>2970419306</PHONE>
                <EMAIL>1802@NetUniversity.org</EMAIL>
                <PHOTO>a1802.gif</PHOTO>
                <ADVISOR_ID></ADVISOR_ID>
```

FIGURE 1 Ipedo console with the UGRADS XML document



Ipedo XML Database Management Console
Database Collection Document Help
Schema View
Document List View
ugrads.xml

Enter Query: NETU/UGRAD[CLASSYR="2002"]

Create Index
Index Key: STATE
Index Type: Equality
Index Implementation: BTree
Index Path: NETU/UGRAD/STATE
Build index now: ☑

FIGURE 2 Building user-defined indexes on documents

API documentation. I found these documents to be well organized and thorough, and programmers will have no trouble working with them. One additional document that I suggest the Ipedo team develop is an integration guide. Conceptually, I could see how to use Ipedo alongside my database and application server, but it would be useful to have some detailed examples to work with. I expect that some of these additional materials will be made available once the Ipedo Technical Support Center is up and running.

Summary

The Ipedo XML database can definitely improve performance for those of you that deploy large XML-based applications. Ipedo's Java API makes it easy to integrate the Ipedo technology into your existing application architectures such as databases, application servers, and edge servers.

I recommend that you look into Ipedo if you're familiar with XML and are looking for a way to improve XML application performance.

[WRITTEN BY JOSEPH SMITH

Encrypting XML documents dynamically

Using the IBM XML Security Suite

Part 1 of 2

With Web services moving to the forefront at a very rapid pace, components will share their data. SOAP, an XML derivative, will be used to exchange data between the services, and XML is the glue that will hold the e-commerce and Web-based solutions together. That powerful glue, however, presents a problem: security.

Purchasing, payment, and even banking will perform functions easily over the Internet using XML. But the XML document needs to be secured just like any other data. This two-part series examines two ways to deal with XML security: encryption and, in Part 2, signatures.

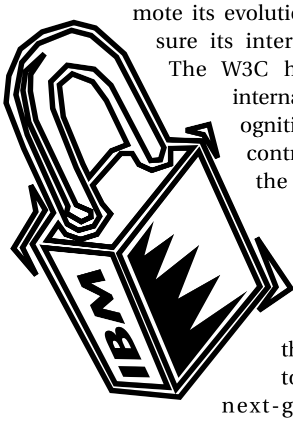
Encryption

There are two levels of XML encryption. One involves encrypting the complete document; the other, a specific element and all child nodes below it.

Any encryption should be a well-known algorithm that should be tested and resistant to known plain-text attacks. Standards such as those promulgated by the World Wide Web Consortium married to well-known algorithms like RSA will create resistance to such attacks.

W3C Working Drafts

The W3C was created to lead the World Wide Web to its full potential by developing common protocols that promote its evolution and ensure its interoperability. The W3C has earned international recognition for its contributions to the growth of the Web. XML 1.0 was introduced as the first step toward the next-generation Web. Since XML 1.0, a number of recommendations have been added, including XML Encryption known as XMLenc.



The goal of the XMLenc Working Group is to develop a process for encrypting/decrypting XML documents and portions thereof. The mission is to provide an XML syntax to represent (1) the encrypted content and (2) the information that enables an intended recipient to decrypt it.

Before an XML document can be secured using the W3C standard, it must be well formed or a valid XML document.

IBM XML Security Suite

IBM has developed an alpha release of XML security tools, the XML Security Suite, which provides security features such as digital signature, element-wise encryption, and access control to Internet business-to-business transactions. Currently, the release contains an experimental implementation of a proposal of the W3C XML Encryption spec. These tools can be downloaded from www.alphaworks.ibm.com/tech/xmlsecuritysuite.

IBM JCE

Sun has placed code in its Java Cryptography Extension that prevents you from using other providers once it's installed. Also, Sun's JCE provides only a few algorithms, and RSA encryption is noticeably lacking – in fact, it's not there. As the IBM JCE provides more algorithms than Sun – mainly the RSA algorithm – this is the JCE I'll use here. You can download it by clicking the WebSphere developer downloads at www.ibm.com/websphere. It's packaged with the IBM JDK.

It's imperative that you follow the instructions for installing any JCE or you'll get some very popular exceptions, such as "NoSuchAlgorithmException," "Algorithm RSA not available," "Cannot set up certs for trusted CAs."

DOM Parsers

I'll use Sun's Java XML parser, JAXP, which is available from Sun at www.javasoft.com/xml.

Setting Up the Keystore

Keytool, an application shipped with the JDK, manages the keystores and can even generate certificates. A *keystore* is a collection of certificates and keys. Depending on the solution, a keystore can be stored in memory, in a file, even in a database. Java uses the java.security.KeyStore package to generate a keystore for memory. Keys and certificates are stored as entries in a keystore and referenced by their alias name. Keystores use passwords to control access to private keys.

You can create your keys statically by using the keytool command found in your {javahome}/bin directory.

An example of creating an RSA key for use in our sample would be as follows:

```
Keytool -genkey -keyalg RSA -storepass
xmlpass -keystore xml.keystore -alias
-mykey
 -dname 'CN=Joseph Smith, OU=
Enterprise Solutions Group,
 O=BMC Software, L=Herndon, ST=Va, C=US'
```

The -idname option specifies an X.500 Distinguished Name. X.500 Distinguished Names identify entities for X.509 certificates.

AUTHOR BIO

Joseph Smith, a senior software engineer with over 13 years of experience, has been designing and developing Web-based soulutions for the last five years.

As you can see, that's quite a long string to have to type in and can be prone to erroneous data. I tend to like the way keytool prompts you if you don't supply all of the necessary information. Try just entering:

```
Keytool -genkey -keyalg RSA -keystore
xml.keystore -alias -mykey
```

A few things to note: if you don't specify a keystore, the default keystore for Windows 2000 will be located at Documents and Settings\{username}\.keystore; on Linux the location might be similar to \home\{username}.keystore.

To retrieve the keystore, get an instance of it using the KeyStore.getInstance(), passing the keystore type as an argument. For now, the default keystore implementation is called "JKS", for Java keystore.

Configuring the Security Suite

Using the IBM Security Suite to encrypt XML documents can be done in multiple ways. For simplicity, I'm going to demonstrate how to encrypt the XML document dynamically. To do this, I create an XML document in memory. This document will contain information needed for encrypting the XML document. This is where things can get hairy, and I strongly recommend you read the W3C Encryption Syntax and Processing documentation, which details how to encrypt data in an XML-conforming manner. More important, it details the elements that will be used in place of the chosen elements for encryption.

Three elements will be used for this example of XML Encryption:

- ***EncryptionData:*** The core element in the syntax
- ***EncryptionMethod:*** The encryption algorithm applied to the data contained in this element
- ***CipherText:*** A mandatory element that provides the encrypted data

Again, I can't stress enough the advisability of reading the W3C Encryption Syntax and Processing documentation. It details these elements and others with regard to various forms of encryption.

With the element-level encryption feature of the IBM XML Security Suite, you can choose which elements of an XML document you want to encrypt. You may encrypt more than one element and still leave the remainder of the XML document intact.

One of the initial steps in the development process is to configure the XML Security Suite. Looking at the javadocs provided with the Security Suite toolkit, you'll see that there are classes for the specific elements nested in the com.ibm.xml.enc package. Included with those classes is an XEncryption class.

Now I need to set up the EncryptionMethod objects to be used and assign the algorithm. The algorithm determines the contents of the node-list within the EncryptionMethod. Algorithm is a mandatory attribute holding a URI identifying it. I'll set up two such objects:

```
encMethodDES = new
EncryptionMethod();
encMethodDES.setAlgorithm('urn:nist-
gov:tripledes-ede-cbc');

encMethodRSA = new
EncryptionMethod();
encMethodRSA.setAlgorithm('urn:rsad-
si-com:rsa-v1.5');
```

The first object sets an EncryptionMethod element to your EncryptedData element because the former element is referenced during the encryption operation so it will know how to encrypt an XML element with a generated key. That EncryptedMethod will use the Triple DES algorithm.

The second object is set up because the key is encrypted according to the algorithm specified in the EncryptionMethod element set to the EncryptedKey element, which will use the RSA algorithm.

Once I create the EncryptedData object, I call the method EncryptedData.setType() to indicate that the encrypted data represents an XML element before encryption. I also want to call EncryptedData.setEncryptionMethod() to set the EncryptionMethod used for the encryption:

```
encData = new EncryptedData();
encData.setType(EncryptedData.ELE-
MENT);
encData.setEncryptionMethod(encMethod
DES);
```

Next comes the simple procedure of creating the KeyInfo object and adding the keyname to it. This object contains a method of identifying the key that was used in the encryption process. *mykey* is the name of the alias we created earlier using the keytool.

```
keyInfo = new KeyInfo();
keyInfo.addKeyName('mykey');
```

The last step in the Security Suite configuration is to set up the Encryp-

> Any **encryption** should be a well-known **algorithm** that should be **tested** and **resistant to** known **plain-text attacks**

tedKey object. The EncryptedKey is used to transport the encryption keys from the originator to a known recipient. To do this, a key transport algorithm is used: RSA, using v1.5 padding. The EncryptedKey object is told about the KeyInfo object and the EncryptionMethod used to encrypt the key:

```
encKey = new EncryptedKey();
encKey.setEncryptionMethod(encMethRSA
);
encKey.setKeyInfo(keyInfo);
```

Because we're constructing an XML document in memory, I need to call createElement for the EncryptedData and the EncryptedKey. These methods will return elements to be used later.

Encrypting the Elements

Once I determine which element I want to encrypt in the XML document, I call the encryptAndReplace method and tell it not only to encrypt the data, but to replace the current element with the encrypted element. This is achieved by calling:

```
XEncryption.encryptAndReplace
(Element elem, boolean contentOnly,
Element encDataElement, Key key,
Element encKey);
```

I'll pass the objects that were configured earlier. Because the attribute of the EncryptionData is "Element", the Security Suite follows the rules of the W3C and replaces the element to be encrypted with an <EncryptedData> element. After encryption, you'll notice two <CipherText> tags: one for the <EncryptedKey>, the other for the <EncryptedData>. The <CipherText> for the <EncryptedKey> is mandatory and contains the encrypted key data. The <CipherText> used for the <EncryptedData> is the data you requested to be encrypted.

Decrypting the Elements

For decrypting we need to focus on the keystore information. To accomplish this, create a KeyInfoResolverforKeyStore object passing in the keystore. After that, the resolver needs the password and alias to determine which key to retrieve from the keystore. Once that's been done, I let an XEncryption object know about the resolver by invoking setKeyInfoResolver:

```
KeyInfoResolverForKeyStore kiRes =
new
KeyInfoResolverForKeyStore(store);
kiRes.put(keyAlias, keyPass);
xEnc.setKeyInfoResolver(kiRes);
```

After that's all done, we want to retrieve a list of all the matched element nodes that use the <EncryptedData> element. I'll use the DomUtil static method getElementsByTagNameNS(). Traverse through the node-list that's passed back from getElementsByTagNameNS() and extract the elements from it. Those elements will be passed to the XEncryption.decryptAndReplace();.

Acknowledgment

I'd like to thank Takeshi Imamura for his help and direction with the XML Security Suite.

References

- ***W3C XML Encryption specification:*** www.w3.org/2000/09/xmldsig, www.w3.org/Encryption/2001/
- ***JSR 106 XML Digital Encryption API:*** http://jcp.org/jsr/detail/106.jsp
- ***Java Cryptography Extension:*** www.javasoft.com/products/jce, www.javasoft.com/products/jce
- ***IBM alphaWorks:*** http://alphaworks.ibm.com

JSMITH@PATRIOT.NET

XML JOURNAL

The World's Leading XML Resource

Home | Advertise | Subscribe | Contact | SYS-CON MEDIA

Dot.com

- buyer's guide
- xml forums
- mailing list
- xml jobs
- xml store

Magazine

- advertise
- authors
- customer service
- editorial board
- subscribe

Content

- archives
- digital edition
- editorial
- features
- interviews
- product reviews
- source code

Conferences

- xml edge
  santa clara, ca
  oct 22–25

XML Training
November 2001

Search XML-J

Search

Check out these companies and their committed XML technology!

XML-J Specials

JDJStore.com
$229.99

JDJ, XML, CFDJ, CFAdvisor - ALL FOUR!!! The Complete Works

Bestsellers

1. Plantronics M-175 Headset for Cellular & Cordless Phones $38.99
2. Garmin: StreetPilot III GPS $829.99
3. Cape Clear: CapeStudio (Single Developer's License) $445.00
4. WebGain Visual Cafe Standard Edition $87.99
5. RoboHELP Office 9.1 from eHelper V8.0 $898.99

What's Online

www.sys-con.com/xml

XML-Journal.com

Visit www.xml-journal.com every day for fast-breaking Java news and events. Stay ahead of the competition by knowing what's happening in the industry, minute by minute.

XMLEdge Highlights

SYS-CON's XMLEdge 2001 International XML Conference & Expo hit the West Coast October 22–25. Top-level XML professionals flocked to the Santa Clara Convention Center for visionary keynotes, information-packed sessions, and new-product demonstrations.

Go to www.sys-con.com/xml for all the show buzz and bustle. See (and hear!) Charles Goldfarb's opening keynote on **SYS-CON Television**. Read the daily show reports for news and commentary. Listen to the energetic Web services CEO panel discussion.

Subscribe to Our Free Weekly Newsletters

Subscribe to Our FREE Weekly Newsletters

□ Java
□ XML
□ Web Services
□ Wireless
□ ColdFusion
□ WebSphere
□ WebLogic
□ Linux

E-Mail Address:

Subscribe

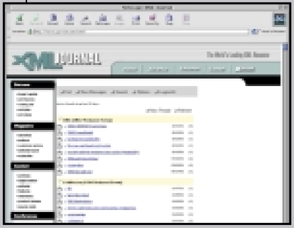
Now you can have the latest industry news delivered to you every week. **SYS-CON** newsletters are the easiest way to keep ahead of the pack. Register for your *free* newsletter today! There's one for Java, XML, Web services, Wireless, and ColdFusion. Choose one, or choose them all!

Reader Feedback

Read our articles online, then critique them! A simple form at the end of each article lets you tell us how we're doing. Did you agree with the author? Does your reseach bring you to a different conclusion? Was there a point you'd like the author to clarify? If you'd like to contact the author directly, you can do that from here, too.

Note: Some feedback may be selected for our print edition.

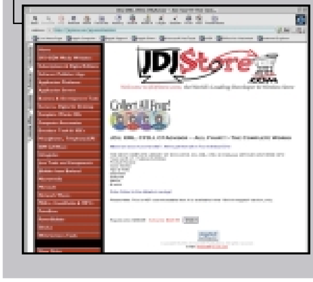
XML Forum



Join XMLJList, a free XML mailing list community. Contact other IT professionals, industry gurus, and ***XML-Journal*** writers for XML discussions, technical questions, and more. Voice your opinions and assessments on topical issues, or hear what others have to say.

Join the XMLJList now to monitor the pulse of the XML industry!

JDJ Store CD Special



The complete library of ***XML-J***, ***JDJ***, ***CFAdvisor***, and ***CFDJ*** articles is available on CD at a special price, for a limited time.

Order now and have over a thousand articles on hand for research and review…features, how-tos, product reviews, case studies, tips & tricks, interviews, IMHOs, and more!

Check out over 250 ***XML-J*** articles covering topics such as XML in Transit, XML B2B, Java & XML, The XML Files, XML & WML, VoiceXML, SYS-CON Radio, XML & XSLT, XML & XSL, XML & XHTML, 2B or Not 2B, XML Industry Insider, XML Script, XML & Business, XML Demystified, XML & E-Commerce, XML Middleware, and much more.

WRITTEN BY ANITA C. NICHOLSON

Harness the full capabilities of XML

RDBMS XML Extraction with SQLJ and Java Stored Procedures

Extracting data from a relational database into a DTD-validated XML file has proved to be an incredibly difficult quandry for the IT/IS/e-commerce industry. While myriad commercial solutions have been offered, many still lack the depth of power and richness of nesting available in XML.

These solutions, ranging from XSQL and object database layovers to generic XML extracts, are laudable in their efforts, yet still contain the following drawbacks:

- ***Simplistic XML:*** XML utilities provided with relational databases typically generate generic <row> or <column> elements.
- ***Document fragments:*** Complete XML documents are rarely returned from base XML queries.
- ***Shallow documents:*** The capability to maintain the hierarchical structure of XML is typically restricted to an empty element tag with an array of attributes or a list of elements lacking attributal depth.
- ***Tightly coupled data:*** While object databases have made leaps in the arena of XML, elements nested to the level of *n* still remain too complex. Additionally, OO databases typically map the DTD directly to the table structure, which makes it difficult to separate the two should the DTD or database structure change.
- ***Noncomplex queries:*** Most XML queries are limited to a single table in a relational database and rarely span multiple data areas.

While not all solutions fall into the points listed above, many still fail to fully coalesce the fluidity of XML with the rigidity of relational databases. The ability to generate XML from your database according to your DTD/Schema to the full breadth and depth allowed in XML1.0 (without limitations) should be easily implementable. Until a commercial solution is available, I propose combining the power of Java stored procedures – Java classes stored in the database and accessed by a normal stored procedure – with SQLJ to generate XML from a database.

What is SQLJ? SQLJ is built atop JDBC. As a result, it can access JDBC connections/statements/resultsets. However, it eliminates the overhead of creating statements in JDBC. Users can execute a SQL query without explicitly calling a java.sql.Connection or java.sql.executeQuery or using java.sql.Statement to create the statement.

So why use SQLJ to generate XML? Let's say you're a developer for BookWorld OnLine. Your users want to retrieve all products written by a certain author from the database in XML. You know you'll have to use the Document Object Model to construct the document. Yet, programmatically, you want a solution that's compact and updatable.

SQLJ's pared-down code leads to easy code maintenance and readability. Using a Java stored procedure to contact a SQLJ class hides the complexity of constructing the XML document from the end user. To demonstrate how to extract XML from a database via SQLJ and Java stored procedures, I'll first give a brief outline of these two Java-based database concepts and then tackle the task at hand: generating XML from the BookWorld OnLine database.

Note: Although I don't promote one parser/IDE/database over another, this solution is based on Oracle's SQLJ utility and uses Oracle 8*i* Personal Edition, JDeveloper 3.2, and XML parsers Xerces 1.3.1 and Jaxp 1.1 on Windows 2000 for development. As a result, my SQLJ solution may focus primarily on Oracle.

Background

While this is an introductory article, the following section will provide a brief background of these Java-based tenets and how to deploy them.

SQLJ

While everyone uses the statement "embedded SQL in Java" to explain SQLJ's functionality, I believe it also offers the following advantages:

- ***Reduced connection worries:*** In SQLJ, unlike JDBC, a default connection is assumed after the connection context's initial declaration. Only if you want to use a different database/schema would a connection context be explicitly stated for each query's execution (for SQLJ default connection see Listing 1, line 31).
- ***Simplified statements:*** No more worrying about the complexity of prepared statements! SQLJ can directly execute a statement, create PL/SQL procedures, and even deploy rollbacks/commits. Utilizing the host variables (:IN, :OUT, :INOUT), you can, respectively, pass values into a query, receive data from it, or declare a variable to both receive and send data (see Listing 1, line 102).
- ***Iterators:*** Iterators are what I believe make SQLJ incredibly powerful. Through strongly typed accessor

AUTHOR BIO

Anita C. Nicholson is an independent consultant who has developed Java and XML applications for over four years. She received her undergraduate degree from the University of Pennsylvania and is a candidate for an MS in computer science from Villanova.

statements, users can use positional (data accessed by column location) or named (data retrieved by user-defined column names) iterators to claim query results (see Listing 1, line 18).

- ***JDBC Access:*** Yes, you can use SQLJ with JDBC. You can cast resultsets to iterators and connection statements to connection contexts. For an in-depth comparison, see Oracle's FAQ (technet.oracle.com/tech/java/sqlj_jdbc).

For more information in general, see the SQLJ tutorial at www.sqlj.org.com.

To run a SQLJ program:

1. Download Oracle's database (SQLJ is included), Oracle's SQLJ utility (www.technet.oracle.com/software), or IBM's SQLJ (www-4.ibm.com/software/data/db2/java/sqlj/).
2. Set up your system environment. You should have the SQLJ.exe (the command-line executable) in your PATH and the following zip files in your CLASSPATH:
   - *Oracle JDBC drivers:* classes12.zip (JDK1.2), classes111.zip (JDK1.1.*)
   - *SQLJ translator:* translator.zip
   - *SQLJ runtime classes:* runtime12.zip
3. Import the following base SQLJ classes into your program:
   - *sqlj.runtime.ref.** (to establish the DefaultContext connection)
   - *Oracle's oracle.sqlj.runtime.** or the *standard sqlj.runtime.** (for runtime environment)
4. Save the document as .sqlj.
5. To compile, type sqlj [filename] at the command line.

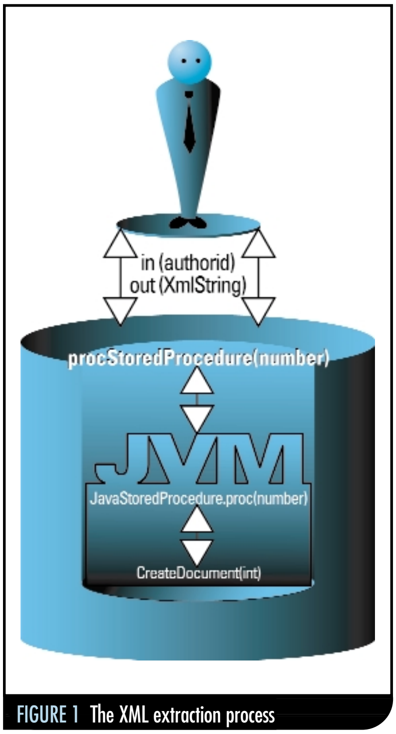


FIGURE 1 The XML extraction process

Java Stored Procedures

If you thought SQLJ was easy, Java stored procedures are even easier. Don't be fooled by the name. In actuality, a Java stored procedure is merely a Java class loaded into the database and accessed by a normal stored procedure. Through this, Java stored procedures offer database functionality without PL/SQL familiarity. Users can access the complexity of Java and embed it in the database as triggers, stored functions, and more.

To create a Java stored procedure:

1. Write a normal Java program. Expose one of the methods to the database. Similar to "main," this must be public and is accessed by the calling SQL procedure (see Listing 2).
2. Compile the program.
3. Load the Java class into the database: type loadjava –user username/password[@connectstring] filename(s) at the command line.
4. Write a stored procedure in SQLPlus (or other) to access the Java class (see Listing 3).
5. To call, type Call [Schema].[procname].

Solution

So, how do we combine the power of SQLJ and Java stored procedures to generate XML from BookWorld OnLine's database? At a minimum you'll need (1) a stored procedure that receives the authorid from the user and is accessible via any client/browser; (2) a Java stored procedure class that's accessible to the database and can access the SQLJ class; (3) a SQLJ class that actually queries the database, constructs the XML Document, and returns the document's String form.

The solution I've provided parallels this three-part paradigm. As illustrated in Figure 1, the user passes the authorid to the stored procedure (procStoredProcedure). The JavaStoredProcedure acts as an intermediary between the stored procedure and the SQLJ class (CreateDocument). It connects the database universe to the Java universe. CreateDocument uses SQLJ to query the database for products related to that author, generates XML via the DOM based on the iterator data, and returns the XML document (Product.xml).

Following is a high-level overview of the code (all line references in this section are to Listing 1):

1. ***CreateDocument():*** Constructs the XML document and utilizes SQLJ to extract the data. Specific explanations of this class's primary parts are:
   - *#sql public iterator ProdIter:* ProdIter

SAVE THE DATE! WIRELESS EDGE MAY 7-9, 2
BUSINESS & TECHNO
wirele
EDGE
WBT speaks with the president of AOL Anywhere
page 18
Going Wireless
DAWN of

contains all product information retrieved from the database. Like all SQLJ iterators, it must be declared prior to use and is allowed only in a class's top level. Like a class/variable, it may also be public/private/static (line 18).

- *Public CreateDocument():* CreateDocument()'s constructor establishes the default connection context to the database via Oracle.connect() (line 31) and the XML Document via DocumentImpl() (line 28).
- *public String getDocString():* Apache's XMLSerializer (line 47) outputs an XML document's String form. Its arguments: StringWriter – enables a user to call the toString() method on the document; and OutputFormat (the way I use it) – declares the XML document, its format (i.e., text), and whether pretty-printing (i.e., indentation) should be used.
- *private Element buildNode():* buildNode(), a helper method (line 59), reduces the tedium of creating an Element, populating the node, and appending it to the document.
- *private static String intToString():* Users who want to set an element's attribute to a row's primary key must convert the ID from a number to a String as the DOM's setAttribute() method only accepts strings (line 78).
- *public void buildDoc():* This constructs the XML document including the root node and all dependent elements. To use ProdIter, you must declare an instance of it (line 101). To execute it, you only need to use #sql and the instance's name (line 102). To extract the data, call the "methods" of ProdIter on its instance (line 112). By allowing the iterator names to mirror the names of the document's node, it's easy to map them to each other and makes your code easier to maintain (line 114).

2. ***ProcJavaStoredProcedure():*** The Java class that deploys CreateDocument() (see Listing 2).
3. ***ProcStoredProcedure():*** The SQL stored procedure exposing JavaStoredProcedure() to the database (see Listing 3). As you can see, the stored procedure SQL input types are mapped to Java's datatypes.
4. ***Product.xml:*** The result the user sees (see Listing 4).

Conclusion

The solution outlined above has the following advantages:

- This temporary architecture lends flexibility should the database or XML structure change.
- A user can consistently generate XML from the database and use XSL to generate any additional formats.
- The combination of Java stored procedures and SQLJ allows simultaneous construction of complex queries and complex XML documents.

For additional information refer to Oracle's Technet. While I believe the commercial solution of joining the power of XML with the popularity of Relational Database Management Systems is just around the corner, the interim combination of SQLJ and Java stored procedures will enable a user to harness the full capabilities of XML without the drawbacks of generic XML, simplified constructs, or tightly bound data objects.

Sources

- Chang, B., et al. (2000). *Oracle XML Handbook*. McGraw Hill.
- Chaudhri, A., and Zicari, R. (2001). *Succeeding with Object Databases: A Practical Look at Today's Implementations with Java and XML*. Wiley.
- Martin, D., et al. (2000). *Professional XML*. Wrox Press.
- Oracle Technet: http://technet.oracle.com.
- Thakkar, M. (2000). *E-Commerce Applications Using Oracle8i and Java from Scratch*. Que.

ANITA.NICHOLSON@VILLANOVA.EDU

LISTING 1

```
/**
  * Returns XML String based on user query
  * @author  A. Nicholson & the Idea proofing staff
  * (Len, Alb, Bor)
*/
import sqlj.runtime.ref.DefaultContext;
import oracle.sqlj.runtime.*;
import java.sql.*;
import java.io.*;
import javax.xml.parsers.*;
import org.w3c.dom.*;
import org.apache.xerces.*;
import org.apache.xml.serialize.*;

public class CreateDocument
```

```
{

  #sql public iterator ProdIter(String isbn,
                                String title,
                                String type,
                                String first,
                                String last);
  Document doc;

  /* ESTABLISH DB CONNECTION & NEW DOCUMENT */
  public CreateDocument()
  {
    doc = new DocumentImpl();
    try
    {
      Oracle.connect("jdbc:oracle:oci8:@","scott",
                     "tiger");
    }
    catch(Exception e)
    {
      e.printStackTrace();
    }
  }

  /* RETRIEVE STRING FORM OF DOCUMENT */
  public String getDocString()
  {
    StringWriter s = new StringWriter();

    try
    {
     XMLSerializer x = new XMLSerializer(s,
     new OutputFormat(doc, "text", true));
     x.serialize(doc);
    }
    catch(Exception e)
    {
      e.printStackTrace();
    }
    return s.toString();
  }

  /* CREATE ELEMENT */
  private Element buildNode(Element parentNode,
                            String nodeName,
                            String nodeValue)
  {
    try
    {
      Element e=doc.createElement(nodeName);
      parentNode.appendChild(e);
      if (nodeValue != null)
e.appendChild(doc.createTextNode(nodeValue));
      return e;
    }
    catch(Exception e)
    {
      e.printStackTrace();
 }
    return null;
  }

  /* RETRIEVE STRING FORM OF INT */
  private static String intToString(int id)
  {
    try
    {
      Integer i = new Integer(id);
      return i.toString();
    }
    catch(Exception e)
    {
  e.printStackTrace();
 }
    return null;
  }

  /* BUILD DOCUMENT */
 public void buildDoc(int inauthorid)
 {
   //Create Root Element
   Element root = doc.createElement("products");
   doc.appendChild(root);

  try
  {
    ProdIter prodIter;
    #sql prodIter =
    {
      SELECT product.isbn,
          product.title, product.type,
          author.first, author.last
      FROM   product, author, productauthor
      WHERE  productauthor.authorid = :inauthorid
      ORDER BY product.title
     };

    while (prodIter.next())
      {
        Element product=buildNode(root,"product",
                                  null);
        product.setAttribute("isbn",
         prodIter.isbn());

        product.setAttribute("type",
         prodIter.type());

        Element title=buildNode(product,"title",
         prodIter.title());

        Element author = buildNode(product,
         "author", null);

        product.setAttribute("authorid",
         intToString(inauthorid));

        Element first = buildNode(author,"first",
         prodIter.first());
        Element last = buildNode(author,"last",
         prodIter.last());
      }
      prodIter.close();
    }
    catch(Exception e)
    {
      e.printStackTrace();
 }
}
```

LISTING 2

```
public class JavaStoredProcedure
{
  public static String proc(int authorid)
  {
   //Call SQLJ Class - Create Document
   CreateDocument createdoc = new CreateDocument();

   //Build Document based on user parameter
   createdoc.buildDoc(authorid);

   //Return String to user
   return createdoc.getDocString();
   }
}
```

LISTING 3

```
CREATE FUNCTION procStoredProcedure(authid NUMBER)
 RETURN VARCHAR2 AS
LANGUAGE JAVA NAME
 'JavaStoredProcedure.proc(java.lang.int)
return java.lang.String';
/
```

LISTING 4

```
<products>
 <product isbn="555-555" type="Java">
  <title>The Java Coders</title>
  <author authorid="1">
   <first>Eric</first>
   <last>Borisow</last>
  </author>
 </product>
 <product isbn="555-657" type="XML">
  <title>The XML World</title>
  <author authorid="1">
   <first>Eric</first>
   <last>Borisow</last>
  </author>
 </product>
</products>
```

DOWNLOAD THE CODE @
www.sys-con.com/xml

BOOK REVIEW

Everything You Need to Know About XML

[Reviewed By Mark Cyzyk]

Author Bio

Mark Cyzyk is the Web architect at Johns Hopkins University in Baltimore, Maryland.

MCYZYK@JHU.EDU

XML: Pocket Reference, 2nd Edition
by Robert Eckstein with Michel Casabianca

Published by O'Reilly and Associates
Price: $12.95

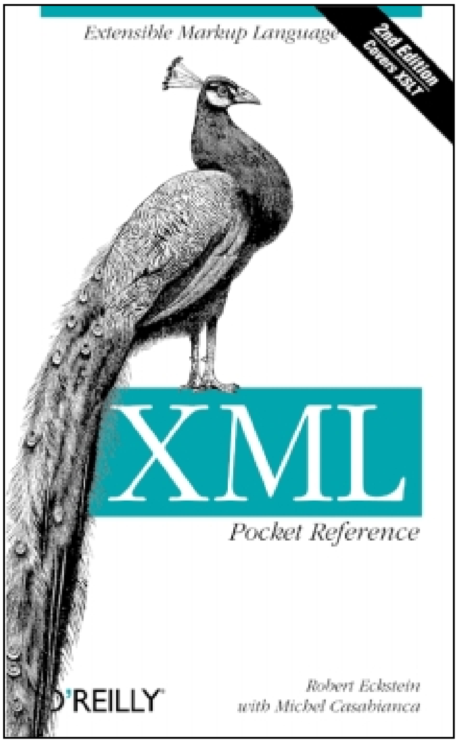
What an excellent little book!

In a scant 96 pages, Eckstein and Casabianca have managed to present everything you need to know to get up and running with XML. After the obligatory review of what XML is and why it's needed, as well as definitions of some of the key concepts of XML technology, the authors launch into a concise, though comprehensive, discussion of DTD (Document Type Definition) design and construction. A DTD specifies the overall structure and content of a valid XML document; it specifies the elements a document can contain as well as the allowed attributes of those elements. Element declarations, entities, and attribute declarations are well covered here, and the examples are clear and unambiguous.

The authors then embark on a discussion of XSL (Extensible Stylesheet Language). The two XML technologies that fall under this rubric are XSLT (Extensible Stylesheet Language Transformations) and XSL-FO (Extensible Stylesheet Language, Formatting Objects). Because XSL-FO is not yet a very mature technology, little attention is paid to it. However, the coverage of XSLT is the richest part of the entire book.

Essentially, XSLT allows you to transform one XML document into another XML format; for example, from a custom XML DTD to XHTML for display in a browser. It converts one set of tags into another set of tags via a mapping in a template file. And XSLT ends up being much more than a way to map tags to tags: the authors illustrate the looping and conditional constructs of XSLT that allow you to program simple logic into the transformation process. After several clear examples of XSLT, the authors provide a long reference section of XSLT elements. It would have been nice if the book contained a long example illustrating the proper use of several of these elements in a stylesheet, but you can get the gist just by looking at their generic, element by element, definitions.

XPATH is the next topic of discussion. XPATH actually underlies the transformation process in XSLT by providing the syntax that locates each node in the document to be transformed. As such, XPATH is sort of a "regular expression" technology of XML. XPATH also provides the direction or "axes" in which the document and its nodes should be traversed, as well as some built-in functions that can be used to do such things as count the number of nodes in a branch of the document, search and compare strings, return the sum of the numerical values contained in a set of nodes, and so on.



So the simple looping and conditional contructs of the XSLT elements, coupled with the rich syntax of XPATH used to locate and otherwise manipulate node values in an XML tree, make XSLT a very powerful tool for data formatting and transformation. Again, one long example illustrating as many of these distinct technologies as possible would have been enlightening.

The last part of the *Pocket Reference* is dedicated to a discussion of XPointer and XLink – technologies that allow for the interlinking of XML documents. As the authors point out, this is essentially the same as the use of anchors and internal links within HTML documents. Use of these technologies basically allows for the linking of one section of a document to another section, either in the same document or in an entirely different one. It's interesting to note that XLink provides for linking from a single link to multiple targets.

With this book O'Reilly has added yet another winner to their fine catalog. As with other O'Reilly publications, the writing style is eminently readable. The typographical conventions used consistently throughout the book make even a quick scan visually informative.

Finally, as the authors note in their introduction: "Some XML-related specifications are still in flux as this book goes to print. However, after reading this book we hope that the components that make up XML will seem a little less foreign."

As far as this reader is concerned, the authors have admirably achieved their objective. I recommend this book highly for those who want to incorporate XML technologies into their Web applications.

XML-J ADVERTISERINDEX

ADVERTISER	URL	PHONE	PAGE
ALTOVA	www.xmlspy.com		19 & 68
CAPE CLEAR	www.capeclear.com/ca30	866-CAPE226	13
DATA MIRROR	www.datamirror.com	800-362-5955	4
JAVA DEVELOPER'S JOURNAL	www.sys-con.com	800-513-7111	54
JDJSTORE.COM	www.jdjstore.com	888-303-JAVA	31, 45
OMNIMARK TECHNOLOGIES	www.omnimark.com	613-745-4242	9
ROGUE WAVE SOFTWARE	www.roguewave.com/guru2		6
SOFTWARE AG	www.softwareagusa.com/xmljweb	877-SAG4XML, ext 545	67
SONIC SOFTWARE	www.sonicsoftware.com/xj		2 & 3
SYS-CON EVENTS	www.sys-con.com	201-802-3069	33
SYS-CON MEDIA	www.sys-con.com	201-802-3020	23
SYS-CON MEDIA, INC. REPRINTS	www.sys-con.com	201-802-3026	39
WEB LOGIC DEVELOPER'S JOURNAL	www.sys-con.com/weblogic	800-513-7111	29
WEB SERVICES JOURNAL	www.sys-con.com/webservices	800-513-7111	58
WIRELESS BUSINESS & TECHNOLOGY	www.sys-con.com	800-513-7111	57
WIRELESS EDGE CONFERENCE & EXPO	www.sys-con.com	201-802-3069	41
XML TRAINING CALENDAR			63
XML TRAINING COURSES			62
XML-JOURNAL	www.sys-con.com	800-513-7111	53

General Conditions: The Publisher reserves the right to refuse any advertising not meeting the standards that are set to protect the high editorial quality of ***XML-Journal***. All advertising is subject to approval by the Publisher. The Publisher assumes no liability for any costs or damages incurred if for any reason the Publisher fails to publish an advertisement. In no event shall the Publisher be liable for any costs or damages in excess of the cost of the advertisement as a result of a mistake in the advertisement or for any other reason. The Advertiser is fully responsible for all financial liability and terms of the contract executed by the agents or agencies who are acting on behalf of the Advertiser. Conditions set in this document (except the rates) are subject to change by the Publisher without notice. No conditions other than those set forth in this "General Conditions Document" shall be binding upon the Publisher. Advertisers (and their agencies) are fully responsible for the content of their advertisements printed in ***XML-Journal***. Advertisements are to be printed at the discretion of the Publisher. This discretion includes the positioning of the advertisement, except for "preferred positions" described in the rate table. Cancellations and changes to advertisements must be made in writing before the closing date. "Publisher" in this "General Conditions Document" refers to SYS-CON Publications, Inc.

www.wbt2.com
SUBSCRIBE NOW
www.javadevelopersjournal.com
TO THE
www.sys-con.com/xml
FINEST
www.coldfusionjournal.com
TECHNICAL
www.sys-con.com/pbdj
JOURNALS
www.webspheredevelopersjournal.com
IN THE
www.wldj.com
INDUSTRY!
www.wsj2.com
subscribe online www.sys-con.com or call 800 513-7111
SYS-CON MEDIA
wireless | java | xml | coldfusion | powerbuilder | websphere | weblogic | web services

Next Month...

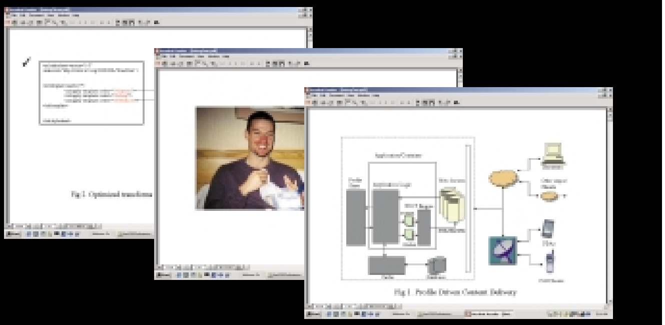
Got XSLT? – Part 3
Transform an XML Example to Speech
by Shouki Souri

XSL Formatting Objects – Here Today, Huge Tomorrow
XSL-FO, while in its early stages, is poised to become the 'next big thing' to present XML data
by Frank Neugebauer

Using the IBM XML Security Suite – Part 2
Encrypting XML documents dynamically
by Joseph Smith

SOAP Messages with Attachments
How the emerging W3C note can be used with the Apache SOAP implementation
by Ian Moraes

Is Your XML Processor Psychic?
Understanding XML namespaces
by David Silverlight



XML JOURNAL

Don't miss the January issue!

XML Training

Crane Softwrights

Lecture and hands-on training materials in XML-related Recommendations can be licensed for both public and in-house use. See http://www.CraneSoftwrights.com/xj/schedall.htm for syllabi and deliveries by G. Ken Holman, http://www.CraneSoftwrights.com/xj/sched.htm for deliveries in Ottawa (including "train the trainer" for licensees), and http://www.CraneSoftwrights.com/xj/licensing.htm for licensees who can deliver Crane's courses at your site.

www.CraneSoftwrights.com/xj
(613) 489-0999

DMSi - Document Management Solutions, Inc.

DMSi presents a full suite of solutions-based training services focused on the specific needs of your environment. Courses include data architecture development, XML, XSL/T and related standards, and product-specific training. We customize these courses to include your data and scenarios related to your organization. The result is a class that satisfies the specific training requirements of your staff.

www.dmsi-world.com
(978) 738-9770

ISOGEN International

Acquire practical XML skills the way the professionals do, with either full track or individual classes based on the ISOGEN XML Learning Framework™. Our pragmatic and highly refined approach, our world-class instructors, and our excellent materials guide you to a higher level of XML skill. Open enrollment, onsite training, and customized courses available.

www.isogen.com or www.datachannel.com
(612) 961-6203

OMNIMARK Technologies

OmniMark Technologies, in conjunction with Stilo Technology in the UK, offers a broad range of XML-related courses covering both design and processing. Courses are offered at OmniMark facilities in Boston, Paris, and Leuven, and at Stiloís facilities in London. Custom and on-site training are also available. For more information see our Web site.

www.omnimark.com or www.stilo.com
(613) 745-4242

Online-Learning.com

Offers online courses for professional development; learn XML, XSL, XSLT, Technical Writing, and Usability. All courses provide extensive instructor support, interaction, practical hands on experience, and certification.

Let us teach you how to deliver single source content.

www.online-learning.com/xmljournal.html or info@online-learning.com (613) 596-5673

TrainingCity

TrainingCity provides XML Consulting services & hands on XML/XSLT training, workshops & elearning. Customized onsite training, CDROMs, & Videos. Call to speak with one of our XML Experts: 858-755-8868, <mailto:info@TrainingCity.com>info@TrainingCity.com, www.TrainingCity.com for course outlines & free online sample training!

www.TrainingCity.com
(858) 755-8868

TrainingCity

For listing opportunites contact
Megan Ring (201) 802-3023 megan@sys-con.com

December 2001

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
3 OmniMark: *XML and Data Transformation,* London XMLmentor: *XML Certification Fast Track Course,* Montreal OmniMark: *XML Programming and Text Processing with OmniMark,* Boston, MA Global Knowledge: *XML Fundamentals,* Chicago, IL Global Knowledge: *XML Fundamentals,* San Francisco	4	5 OmniMark: *XML Processing with OmniMark,* London DMSi: *Introduction to DTD Alternatives,* Classes onsite	6 OmniMark: *OmniMark in Depth,* Boston, MA	7	8	9
10 OmniMark: *Converting to XML Using OmniMark,* London Global Knowledge: *XML Fundamentals,* Boston, MA Global Knowledge: *XML Fundamentals,* Raleigh, NC AppDev: *XML & XSLT: Level 1,* San Francisco, CA Online-Learning.com: *Professional Technical Writing, Introduction to XML, Professional XML Authoring, XSL Techniques, Usability Testing, Information Design Forum* Classes available online or on CD	11	12 AppDev: *XML & XSLT: Level 2,* San Francisco, CA OmniMark: *XML Schema Message Development Workshop,* Boston	13 OmniMark: *XSLT Masterclass,* Boston, MA	14	15 OmniMark: *Data Modelling Masterclass,* Boston, MA	16
17 OmniMark: *Web Programming with OmniMark,* London DMSi: *Introduction to XLink and XPointer,* Classes onsite AppDev: *XML & XSLT: Level 1,* Minneapolis, MN	18	19 AppDev: *XML & XSLT: Level 2,* Minneapolis, MN OmniMark: *Integrating Information with OmniMark,* London	20	21	22	23
24	25	26	27	28	29	30

Happy Holidays!

🟥 AppDev 🟧 DMSi 🟨 Global Knowledge 🟩 OmniMark Technologies 🟦 Online-Learning.com 🟪 XMLMentor

XML News

Agital Enhances Solution Development System

(Bothell, WA) – Version 1.5 of the Agital Solution Development System is now available, according to Agital, Inc., a provider of data conditioning solutions. This release provides value-added enhancements including improvements to guaranteed messaging, failover, and usability features.
www.agital.net

NeoCore Offers Trial Version of XMS

(Colorado Springs, CO) – NeoCore is offering a downloadable trial of its product, NeoCore XML Information Management System (XMS). The XMS Evaluation PAK, available on NeoCore's Web site, demonstrates the speed and ease with which developers can build, change, and create applications around a complex, native XML database.

The PAK supports two simultaneous sessions and the ability to build an information store of up to 10MB. The trial version includes all of the documentation needed for developers to begin building a working application.
www.neocore.com/evaluation

OASIS Acts to Define XML Document Library

(Boston) – OASIS members have formed the OASIS Universal Business Language (UBL) Technical Committee to define a common XML business document library. UBL will provide a set of XML building blocks and a framework that will enable trading partners to unambiguously identify and exchange business documents in specific contexts. Uniting disparate efforts underway by companies and standards groups around the world, the committee intends to enhance and harmonize overlapping XML business libraries and similar technologies to advance consensus on an international standard.

Initial members of the OASIS committee include Arbortext, Boeing, Commerce One, SAP, SoftQuad Software, Sun Microsystems, VerticalNet, Vitria, XML Global, and other OASIS contributors and individual members.
www.oasis-open.org

Cape Clear Ships New Versions of XML Web Services Products

(Campbell, CA) – Cape Clear Software has released new versions of its Web services products CapeConnect and CapeStudio. These versions reportedly add extensive functionality to the product range with wider platform support, enhanced development tools, and full UDDI functionality.
www.capeclear.com

W3C Recommends XSL 1.0

(Cambridge, MA) – The World Wide Web Consortium has issued the Extensible Stylesheet Language (XSL) 1.0 as a W3C Recommendation, representing cross-industry agreement on an XML-based language that specifies how XML documents may be formatted.

W3C Recommendation status indicates that a specification is stable, contributes to Web interoperability, and has been reviewed by the W3C membership, who favor its adoption.
www.w3.org

NeuArchitect 3.5 Available from NeuVis Software

(Shelton, CT) – NeuVis Software has announced the availability of NeuArchitect 3.5, the cornerstone of the NeuVis e-business development platform.

The release features new modeling, construction, and integration capabilities that claim to ease and accelerate development of end-to-end e-business applications that use XML and Web services.
www.neuvis.com

XML Solution for Mortgage Legacy Systems Unveiled

(Ponte Vedra Beach, FL) – Decade Systems Corp. has launched Total Enterprise Data Integration 4.1. TEDI 4.1 purportedly enables lenders to convert data both to and from XML for all mortgage transaction types, making it easier for everyone involved in the loan process to share information. Additional features include HTTP/HTTPS support, expanded analyst functions, enhanced relational database support, and improved communications and report options.
www.decadesystems.com

Nimble Technology Releases XML-Based Software Platform

(Seattle) – Nimble Technology, Inc., is shipping Nimble Integration Suite 1.0, an XML-based data integration software platform for Web services and applications. The suite can query and integrate data from multiple disparate data sources to provide a unified result.
www.nimble.com

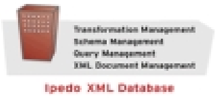
Allora 2.5 Web Service Delivers Internet RDB Access

(San Jose, CA) – HiT Software has released Allora 2.5, which features a system-level Web service for accessing relational databases. Allora 2.5 consists of a graphical Mapper application, runtime server-side Java classes, and client interfaces. It can be installed on any JDK 1.3 platform and supports all major relational databases via JDBC 2.0 SQL middleware.
www.hitsw.com

SYS-CON Show Venue for Ipedo Announcement

(Santa Clara, CA) – Ipedo, Inc., released the latest version of the Ipedo XML Database at the XMLEdge 2001 International XML Conference & Expo in October (see **Product Review**, this issue).

The Ipedo XML Database Version 2.0 has a number of new features, including free-form XML search, scalable vector graphics management, distributed database management, integrated XSL transformation, and large document processing.
www.ipedo.com



FileNET, Arbortext to Deliver Customized Documents

(Costa Mesa, CA) – FileNET Corp. and Arbortext are working to combine the former's Panagon strengths in process management and enterprise content management services with Arbortext's XML-based authoring, conversion, and multichannel publishing of customized documents.

With the integration, customers will be able to create content once and repackage and deliver it for multiple audiences and multiple types of media.
www.arbortext.com
www.filenet.com

Brainhat Unveils New Server

(East Hartford, CT) – Brainhat Corporation has developed the Brainhat Voice XML/Web Server, which interprets natural language requests and provides dynamic delivery of content, enhancing

XML News

user interactions with Web sites and changing the Web programming paradigm. The server is built on the Brainhat operating system, a natural language OS that can be programmed in English.

The Brainhat OS supports Linux and Windows 98/NT/2000. It comes with clients that support SAPI and other third-party APIs, and it supports VoiceXML.
www.brainhat.com

DataMirror Debuts New Server at XMLEdge 2001 Expo

(Santa Clara, CA) – Among other companies introducing new products at **SYS-CON Media**'s XMLEdge 2001 International XML Conference & Expo in late October was DataMirror Corporation. The company's Transformation Server for XML is the latest addition to their Transformation Server software.

The server includes breakthrough real-time, event-driven data capture and event definition to enable businesses to capture selected business events – such as the posting of purchase orders or issuing of invoices from any application database – and send it in industry-standard XML formats across the enterprise and beyond.
www.datamirror.com

eXcelon Releases New Platform, Business Excelerators

(Burlington, MA) – eXcelon Corporation, provider of industry-focused technology and solutions that power the extensible enterprise, has released its new XML-based platform and the first three Business Excelerators of its eXtension Suite. The Excelerators are designed for the insurance, chemical, and manufacturing industries. Each has at its core eXcelon's service-based architecture.

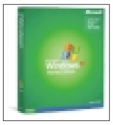
The eXcelon XML platform provides a high-performance, low-risk option for organizations seeking to leverage existing technology investments and eliminate problems associated with adjusting applications to accommodate new business requirements.

The eXcelon Business Excelerators, which run on top of the platform, are workflows, services, and common information models designed to support industry-specific business requirements. Both offerings are designed to enable corporate developers to create customized applications by allowing for quick and flexible integration and the automation of ever-changing business processes.
www.exceloncorp.com

Gates Takes Wraps Off Windows XP

(New York) – Microsoft has officially launched Windows XP, the newest version of its operating system. Microsoft, chipmaker Intel, and PC makers are expected to spend a combined total of more than $1 billion on marketing for Windows XP.

The operating system ushers in new features tied to Microsoft's long-term strategic plans for media player software, digital photo tools, and online services. It comes in two versions: Home and Professional. Although they appear identical, the latter allegedly offers more sophisticated networking, better security, and support for multiple processors.



Some other highlights of Windows XP: backwards compatibility, better text for those using LCD monitors, multiple desktops, better drivers, stronger security, digital imaging.
www.microsoft.com

Software AG Offers XML-Based Integration Solutions

(San Francisco) – Software AG, Inc., U.S. subsidiary of Software AG, Europe's largest system software provider, has launched a reconfigured and more robust suite of EntireX brand products, its next-generation XML-based integration server. Software AG is a leader in cutting-edge technology for data management and electronic business and the provider of the award-winning Tamino XML Server.

EntireX is a family of flexible and easy-to-use component-based solutions for integrating enterprise applications across heterogeneous operating systems, applications, and platforms – a fundamental element for enabling e-business. EntireX provides integration up to 10 times faster than traditional technology and eliminates up to 80% of traditional coding with its graphical point-and-click functionality.
www.softwareagusa.com

SilverStream Releases XML Integration Engine

(Billerica, MA) – SilverStream Software, Inc., is now shipping SilverStream eXtend Composer 3.0, a fully standards-based XML integration engine that enables organizations to dynamically access a broad range of enterprise information systems using powerful Web services.

Some of eXtend Composer's key benefits include rapid time-to-market, existing enterprise systems integration, and support for Java, J2EE, XML, and Web services.
www.silverstream.com

OASIS Committee to Define Framework for Exchanging Info

(Boston) – OASIS has announced that its members have formed the OASIS Provisioning Services Technical Committee to define an XML-based framework for exchanging user, resource, and service provisioning information. The committee will collaborate to develop the Provisioning Services Markup Language (PSML), an end-to-end specification for the automation of user or system access and entitlement rights to electronic services.

Organizations collaborating on the committee include Access360, Business Layers, Jamcracker, Novell, Oblix, OpenNetwork Technologies, and other OASIS contributors and individual members. Participation is open to all OASIS members; for information on joining OASIS visit www.oasis-open.org/join.

enumerate Unveils Web Publishing Software

(McLean, VA) – Software developer enumerate Solutions Inc. has released Web Developer Toolkit, a first-of-its-kind software package that enables Web developers to create and post "interactive numbers" on Web sites or intranets in the form of embedded applets.

Unlike sites that present numbers in a static format, visitors to sites using enumerate's technology will be able to create custom charts and tables with a simple mouse click. They'll be able to add or remove data from charts to explore correlations and "what if" scenarios, apply one-click mathematical equations to data, and share analyses with colleagues without having to transfer the data to a spreadsheet program.
www.enumerate.com

These Requests Are Really Starting to Get Weird

You sickos have got to be stopped

Kris Kringle's history is kind of a mystery. He didn't seem too interested in giving us much information on his technical experience. We know that he started working in Russia, so maybe he knows Cobol or something. Regardless, unlike normal celebrities he is a wise and innocent sage. We are excited that such a personality would want to write for us, so we give him the floor.

WRITTEN BY **SANTA CLAUS**

You pigs. You absolute sickos. What have you done to the youths of this world? They ask for unholy things that I can't possibly deliver. And I know you're behind this.

Just what the heck is XP? Sure, there's a generation gap I have to deal with every once in a while. New gift ideas take me by surprise. VCRs, DVDs, Apple computers, the list goes on. I adjust. I deal. I eventually get all the kids the toys they want. But when I start getting wishes for complete lists of serial cracks, I know you computer geeks are up to something.

I work around the clock to keep up with the wishes of the young children. But there are a few things that you do to them that make my life difficult. Maybe it's the way you Peter Pans pit your petty ideologies against each other, then get the children involved. When I get letters saying things like, "Please don't make my Daddy's iMac programs into Microsoft .NET files that point to Win32-specific components," I know you've been messing with their heads again.

One child wanted to "be ph34red by all Quake III clans." One wanted me to give him a "client-to-client protocol program that worked smoothly." Give me a break, please. Or a gun.

And that's another thing. Kids used to want toy laser guns. Now they want rail guns. Huh? One boy wanted "the Half-Life Counterstrike version of the rail gun. You know, the one where you can shoot the bad guy in the foot and you still get a frag out of it." No, I'm afraid I'm not familiar with that one.

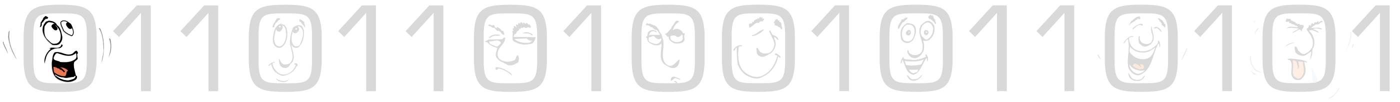
And just what is a "frag"? Children want this more than ever. They want to have more of these than any other child has. What is it? A novelty frog toy of some sort? I can't even find these things on eBay.

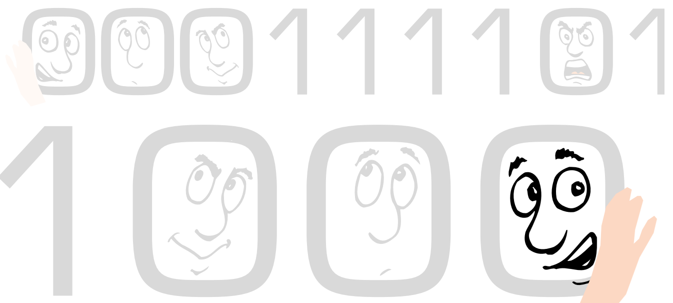
While I'm on the subject of gripes, the "Annual NORAD Tracks Santa Claus" project (at www.noradsanta.org) makes me a tad uncomfortable. You computer addicts don't actually track me with missile defense systems, do you? If you continue to do so, I know one naughty computer worker who won't be getting a bagful of frags this year.

Have you been breeding unnatural animals with your wretched science? I've had children ask me for nearly every breed of dog. But I've never once heard of the SONY breed of dogs. Apparently you've enhanced dogs with some sort of blasted metalwork and electric sensors. Forget coal in your stockings. You'll pay for this in hell, I tell you. And thanks so much for making the children ask me for these things.

Are these SONY people also responsible for those unholy "Poh-kee-mon"? For years you tech people have encouraged the development of "bioengineering" and other distasteful technologies. That and your love of mass industrialization have resulted in "Poh-kee-mon," these electrically destructive mutant mice and humanoid berserkers. Darned if I'll put these former denizens of the inferno into my sac and fly with them at 5,000 feet!

There are alternatives to exhausting me like this. Can't you go back to teaching kids I don't exist or something?





WRITTEN BY **TOD EMKO**

Tod Emko wrote humorous articles for the Syracuse Herald-Journal and held various writer, editor, and cartoonist positions at other publications before accepting his role as a computer nerd. He has years of experience as a front-end Web developer and Perl programmer and is now a senior XSL script architect and XML documentation writer for HotJobs.com.

TEMKO@TOASTEDPIXEL.COM